(EN-P1091)

電腦時代 COMPUTING AGE

22

工商業●教育●家庭娛樂●私人應用電腦雜誌　港幣十元

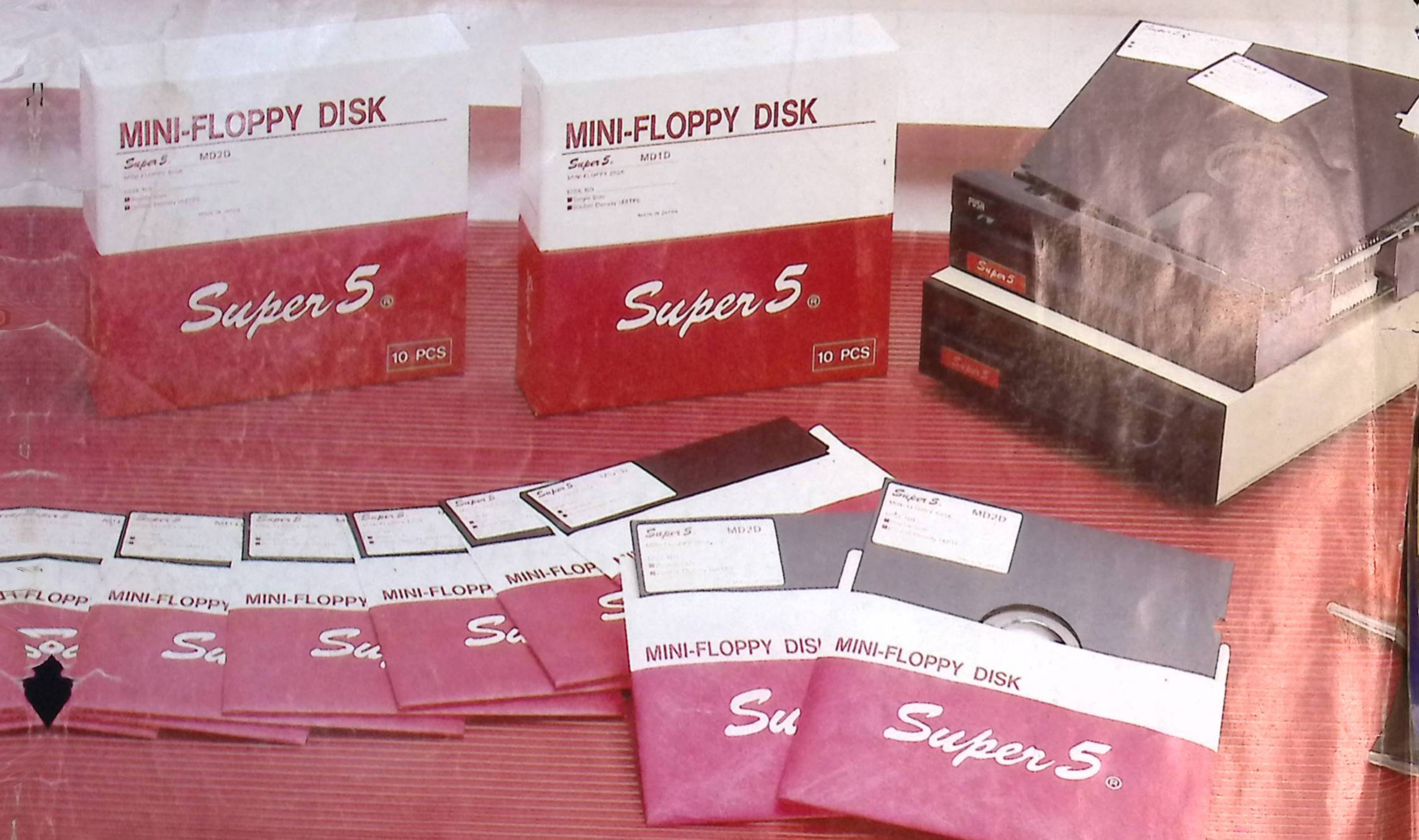


CE彩色電腦磁碟，是當代最先進的產品，令人在單調的電腦操作中，耳目一新，平添不少彩色繽紛的生氣，誠爲電腦用家的知己良伴。
- ……磁碟特有的彩色保護套，不留指印，既能防塵侵染，又可免磨擦生電。除顏色鮮艷外，更易於分門別類，以適合各用家的不同須要。
- CE彩色電腦……性幼細平滑，並經特製之潤滑塗膜處理，可抵禦長期的磨擦，使碟內的資料得以保存無損。在適當的操作下可承受 100,000,000次以上的重覆……讀。
- CE彩色電腦磁碟表層磁……具抗熱和防震功能，不會出現彎曲和摺疊現象，可說是「全天候」磁碟。保證在正常環境下操作，寫入資料與讀出時 100%無誤差。
- CE彩色電腦磁碟比其他產品更……

香港代理：華益企業公司
NO. 31-33 KING'S ROAD, 1ST FLOOR, (NEAR LAU LI ST.) NORTH POINT. G. P. O. BOX 10617, HONG KONG
TELEPHONE: 5-784928, 5-783129
CABLE ADDRESS: "PETCOMY" HONG KONG
……PECO HX
BANKER: HANG LUNG BANK LTD.

MD2D MINI-FLOPPY DISK

資宝牌 軟磁盤



* High quality mini-floppy disk for data storage and retrieval.
* 100% error-free of disk surface ensured by comprehensive tests.
* Prolongs drive read/write heads by its superior surface smoothness.
* Best selection of jacket optimize drive characteristic.

Item Model No.	MD1D	MD2D
Disk Size (Inch)	5¼	5¼
Sector	Soft	Soft
Usable Side	Single	Double
Density	Double	Double
Tracks per inch (TPI)	48	48
Number of Tracks	40	80
Storage Capacity (Bytes)	163,840	327,680

Super 5

A & N ENTERPRISE CO., LTD.
20/F WESTERN CENTRE.,
40-50 DES VOEUX ROAD, WEST, H.K.
TEL: 5-461179 5-488463
TELEX: 65218 MCLWW HX
FAX: 5-497149

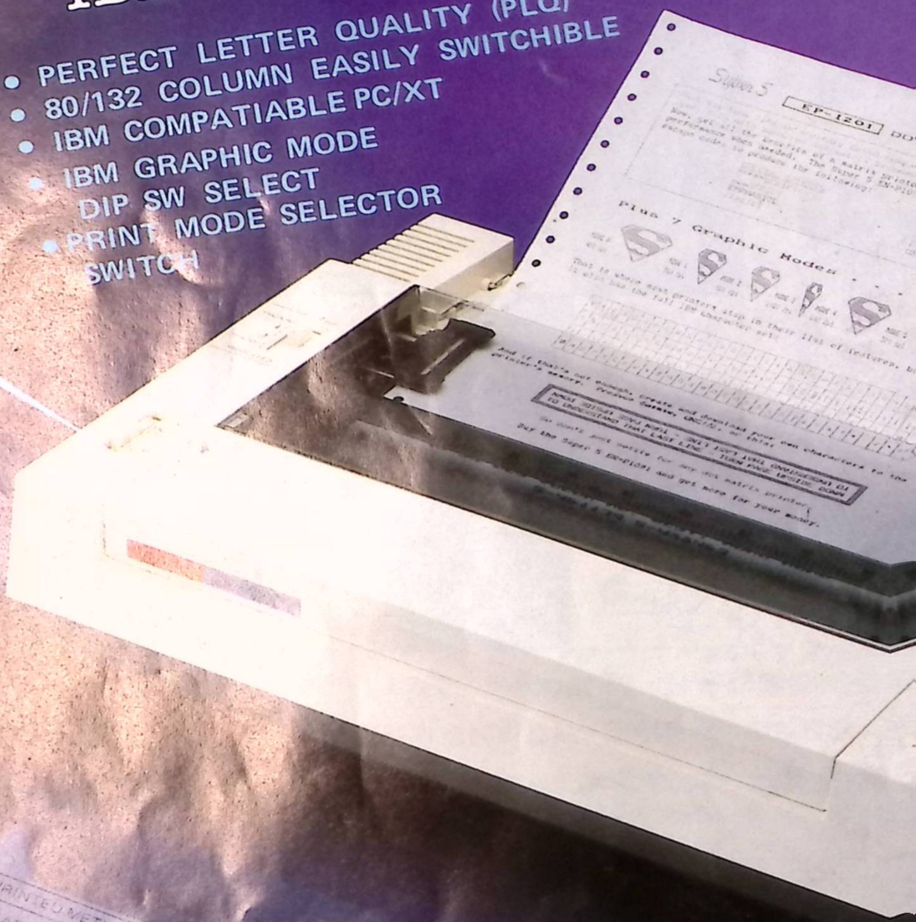
Super 5 EP-1201

A TWO-IN-ONE PR

IDEAL FOR PERSONAL & BU

- PERFECT LETTER QUALITY (PLQ)
- 80/132 COLUMN EASILY SWITCHIBLE
- IBM COMPATIABLE PC/XT
- IBM GRAPHIC MODE DIP SW SELECT
- PRINT MODE SELECTOR SWITCH

Std
Pgm.
L.Q.
Prop.
PRINT MODE



		EN-P 1090
PRINTED METHOD	Serial Impact dot matrix	
HEAD SIZE	9 pin	
PRINT MODES	Draft (pica, elite), Near-letter quality, Proportional, Graphics	
PRINT MODE SPEED (CPS)	Draft 120; Near-letter Quality 29; Proportional 75	pica 80 cps; elite 96 cps
PRINTING CHARACTERISTIC		
Character matrix	9 x 9	9 x 9
IBM PC Compatible	132 IBM PC special characters; 64 block graphics	
Character set	Full 96 characters ASCII with descenders; 32 International characters, (11 countries); Italic	Full 96 character ASCII with descenders
PRINTING SIZES	CPI Pica/Elite/Compressed; CPL Pica/Elite/Compressed	Selectable by DIP SW or programmable
Draft	CPI 10 12 17; CPL 80 96 132	Characters per inch: Pica 10, Elite 12; Characters per line: Pica 80, Elite 96
Block graphics	CPI 5 6 8.5; CPL 40 48 66	5, 6; 40, 48
Near-letter quality		14, 17; 113, 136
PAPER FEED	Adj. tractor feed and friction feed	
INTERFACES	8 bit parallel	
BUFFER	1K text buffer	
OPTIONAL	RS232C 4K buffer board	

香港教育司署立案（1957年創校）

世界函授學校

規模大　優點多　科目好

講義
全部經教育司署審定，閱讀容易，印刷精美，免費供應。



導師
國內外著名大學學士，經香港教育司考核合格，教學認真，決不馬虎。



校址：
香港九龍彌敦道五六六號9樓A座
電話：③—803532

實用商科
著重實際應用，加强商業學識，期間短，學習易，六個月畢業。



實用數理
可以加强數理基礎，費用少，收效大，6個月畢業。

x+y=z

半導體裝修科
配合進步的技術，具初學或職業的裝修人員必修的學科，十個月畢業。



初級無綫電科
全部圖解教導，原理裝修並重，極易學習，十個月畢業。



初級電腦科
電腦基本原理，電腦操作，邏輯電路，程式編寫，八個月畢業。



文憑
本校頒發文憑，均經政府簽証，實是最佳學歷証件。



費用
除低廉之學費外，學生決無額外負擔。



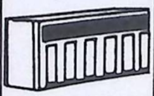
應用漫畫
漫畫可訓練頭腦，增强思想，取材新，學習易，六個月畢業。



商業美術科
內容新，範圍廣，包括美術和最新廣告設計。一年畢業。



冷氣工程
冷氣，冷藏人員缺乏，專門學校極少，一年半畢業。



汽車工程
適合職業及業餘人員加强對汽車方面之學識。



無綫電工程
設計，研究和深造的理想學科。一年半畢業。



本港學生可利用電話詢問索取章程　電話③－803532

請依綫剪下，填好姓名，回郵地址，貼在信封上投寄，本校即免費寄上簡章

郵件請寄：P.O. BOX 70011, KOWLOON, CENTRAL POST OFFICE
九龍中央郵政局信箱 70011 號

技術就是財產

（寄自）

姓名：

地址：

（請註明要學的科目）

- ☐初級電腦科
- ☐無綫電工程科
- ☐初級無綫電科
- ☐半導體裝修科
- ☐實用數理科
- ☐汽車工程
- ☐冷氣工程
- ☐實用商科
- ☐商業美術
- ☐應用漫畫

請貼上郵票

香港教育司署立案

To：世界函授學校註册處

P. O. BOX No. 70011
KOWLOON CENTRAL POST OFFICE,
HONG KONG.

香港九龍中央郵局信箱70011號

電話：K-803532

學問就是力量

暑期超值大貢獻 訂閱電腦時代可免費獲得

三本新出版技術叢書及「微電腦技術」雙月刊十二期

凡於一九八五年九月十五日之前訂閱「電腦時代」，將可獲多項贈品及半價訂閱「微電腦技術」優待。

■ 訂閱「電腦時代」＋「程式磁碟」十二期，可得下列任何一項贈品：

A　免費贈閱「微電腦技術」雙月刊十二期

或 B　免費得到本出版社最新出版之三本電腦技術叢刊（GPLE 使用方法詳析，128 K RAM 咭應用技術詳析 及 MOCKINGBOARD 技術應用）

或 C　免費獲得高級 1S／2D 磁碟一盒（十張）

■ 訂閱「電腦時代」雜誌十二期，可獲得半價優待訂閱「微電腦技術」雙月刊及半價優待購買下列任何出版叢書：

1. GPLE 使用方法詳析
2. 128 K RAM 咭應用技術詳析
3. MOCKINGBOARD 技術應用

閣下可選擇利用郵付／或親身到電腦時代讀者服務部領取贈品或半價購買各類新書。同時，閣下所訂閱之雜誌、磁碟可採用郵寄或每月親身到讀者服務部方式領取。歡迎現時長期訂戶延長訂期。如是長期訂戶請填上原有之訂戶號碼。

所有利用郵寄之贈品將會以掛號形式寄到訂戶府上，毋須加付郵費。

PROFESSIONAL COMPUTER CENTRE
Tel: 3-328248, 3-318960

☐ 請即接受本人訂閱「電腦時代」雜誌連程式磁碟十二期，由第____期開始。我選擇〔　〕（請填上A、B或C）項贈品。麻煩貴社用掛號方式將贈品寄來舍下，我毋須加付郵費。

☐ 請即接受本人訂閱「電腦時代」雜誌十二期，由第____期開始。同時本人有權用半價訂閱「微電腦技術」雙月刊及半價購買新出版之三本技術叢書。

			（請✓）
訂閱	電腦時代＋程式磁碟十二期	$ 475	（　）
訂閱	電腦時代雜誌十二期	$ 120	（　）
半價購買	GPLE 使用方法詳析	$ 15	（　）
	128 K RAM 咭應用	$ 15	（　）
	MOCKINGBOARD 應用	$ 15	（　）
半價訂閱	微電腦技術雙月刊十二期	$ 60	（　）
	合共：	$	

讀者可利用影印本寄來，訂費可用劃線支票或郵政滙票（請勿劃線）或親自到讀者服務部交付現金。支票收款人請填寫「電腦時代出版社」。郵寄地址是：九龍中央郵局郵政信箱 71193 號。

支票號碼＃ ____________　銀行＃ ____________

欲罷不能　讀者反應熱烈
延長優待期至十月十五日

以下為本人的通訊地址（請用英文正楷書寫）：

姓名：____________　電話：____________

地址：____________

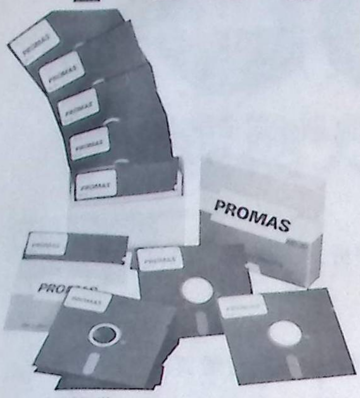
同時請填好下列三個之回郵地址：

姓名：____________	姓名：____________	姓名：____________
地址：____________	地址：____________	地址：____________

本人已是長期訂戶，訂戶號碼是：____________

☐請利用郵寄方式將贈品及雜誌寄來舍下。
☐本人將每月親身到讀者服務部取雜誌及贈品。

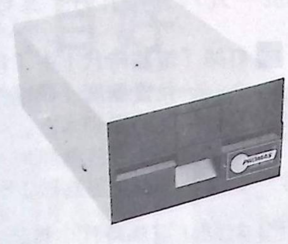
PROMAS 電腦產品



5.25″磁碟
雙面雙倍密度，儲存量達500 K，適合一般電腦使用。全日本製造。



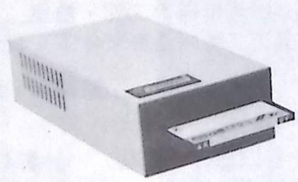
MODEL JT 5.25″薄身磁碟機
多種「WRITE PROTECT」及「HEAD SEEK SPEED」選擇，適合不同應用。APPLE II*及相容電腦專用。全日本製造



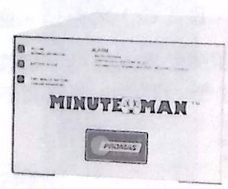
AS-100 5.25″磁碟機
傳統設計，品質檢定合格，APPLE II*及相容電腦專用。全英國製造。



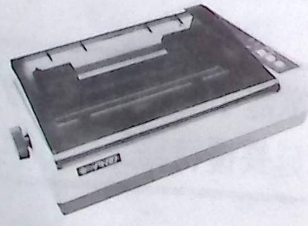
MODEL JA 5.25″磁碟機
全新設計機械部份，特別柔順。APPLE II*及相容電腦專用。全日本製造。



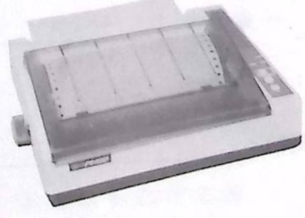
305 3″磁碟機
多重保護資料，與一般APPLE II* 磁碟機共用或可單獨使用。全日本製造。



MINUTEMAN後備電源供應器
防止一般電源問題對電腦及精密儀器造成之破壞。全日本製造。



PX-80印字機
印字速度每秒80字。CENTRONICS介面接口，適合一般電腦使用。全日本製造。



PX-100印字機
印字速度每秒100字，功能與PX-80相同。全日本製造。



PF-150 手提圖文傳真機
攜帶方便，可與世界各地傳真機互傳訊息。全日本製造。

總代理：

PROMAS COMPUTER CENTRE LTD.

寶馬士電腦有限公司

九龍尖沙咀柯士甸路22—26號好兆年行 608 室 電話：3-674687 電訊：HX 38316



PROMAS 5.25"磁碟
100% Error Free

- 鋁合金HUG RING。
- 碟身採用「FERROUS-OXIDE COATING」，令寶貴資料不受環境影響而消失。
- 特製POLYPROPYLENE保護膠盒，容易整理。
- 全日本製造及品質檢定合格。

OPERATIONAL PERFORMANCE

MODEL	Sides	Density	Sector	Track	TPI	Format	Capacity
MD/2D	2s	Double	Soft	35 or 40	48	None	500KBytes

總代理：

PROMAS COMPUTER CENTRE LTD.

寶馬士電腦有限公司

九龍尖沙咀柯士甸路22－26號好兆年行608室 電話：3-674687 電訊：HX38316

體積超小

RITEMAN「威猛牌」印字機系列

適合閣下不同需求！

總代理：

PROMAS COMPUTER CENTRE LTD.

寶馬士電腦有限公司

九龍尖沙咀柯士甸路22－26號好兆年行 608 室　電話：3-674687　電訊：HX 38316



印字特快

- 超薄機身，體積爲普通印字機一半。
- 印字速度特快，每秒達160字。
- 多種印字功能，可印多國文字，更可作BIT IMAGE GRAPHIC DUMP(中國文字及圖案)。
- 可用軟件及硬件選擇「高質素」(LETTER QUALITY)字體，適用於文字處理。
- 採用耐用之盒式絲質墨帶，用戶可補充油墨重覆使用，操作簡易，墨帶壽命高達一百萬字，減少更換墨帶次數，不斷爲你節省金錢。

	RITEMAN PLUS	RITEMAN BLUE PLUS	RITEMAN II	RITEMAN 15
Speed(Character Per Second)	120	140	160	160
Paper Feed				
Friction Feed	●	●	●	●
Pin Feed	●	●	●	●
Tractor Feed	OPTIONAL	OPTIONAL	OPTIONAL	●
Programmable Page Length	●	●	●	●
Programmable Page Brake	●	●	●	●
Programmable Line Spacing	●	●	●	●
1/6"	●	●	●	●
1/8"	●	●	●	●
7/72"	●	●	●	●
n/72"	●	●	●	●
n/216"	●	●	●	●
Programmable Character Styles				
Combinations	66	128	128	128
Pica	●	●	●	●
Elite		●	●	●
Condensed	●	●	●	●
Expanded	●	●	●	●
Emphasized	●	●	●	●
Double Strike	●	●	●	●
Underline	●	●	●	●
Super/Subscripts	●	●	●	●
Italics	●	●	●	●
Proportional Spacing			●	●
Download Characters			●	●
Character Sets				
IBM Graphic Characters		●	●	●
American	●	●	●	●
French	●	●	●	●
German	●	●	●	●
British	●	●	●	●
Danish		●	●	●
Swedish		●	●	●
Italian		●	●	●
Spanish		●	●	●
Graphic Modes				
60 Dots/Inch	●	●	●	●
72 Dots/Inch	●		●	●
80 Dots/Inch		●	●	●
90 Dots/Inch		●	●	●
120 Dots/Inch	●	●	●	●
240 Dots/Inch		●	●	●
Quiet Mode		●	●	●
Interfaces				
Centronics Parallel	●	●	●	●
RS-232C Serial	OPTIONAL	OPTIONAL	OPTIONAL	OPTIONAL
Warranty	1 YEAR	1 YEAR	1 YEAR	1 YEAR

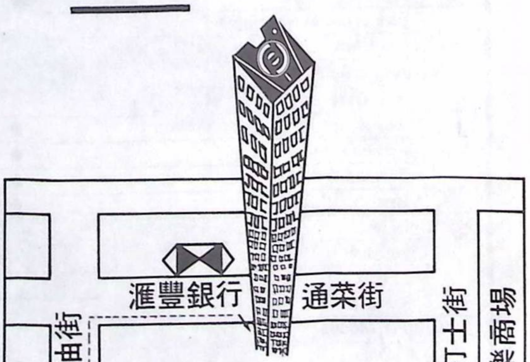
何必自己動手

免費 由專業人員上門爲你修理微型電腦



請即加入本公司之保養及免費維修服務

24小時熱線電話服務
3-848129



本公司現已搬往旺角通菜街1A-1J 威達商業大廈19/F，1905室 TEL：3-848129

微型電腦保養維修中心
ROM & RAM computer ltd

OFFICE: Rm. 1905 Witty Coml. Bldg., 1A–1J, Tung Choi St., Kowloon. TEL：3-848129
Telex: 57551 ROMHX

歷史悠久　深受推崇

16 BIT 電腦專業課程

隨着電腦的日漸進步，16 Bit 電腦愈趨普遍。專業電腦中心爲配合社會的需要，特設立多項16 Bit電腦專業課程，培養電腦人材。

各項 16 Bit 電腦課程：

(1) 16 Bit 電腦基本操作訓練

(2) 資料處理訓練 (DBase III/II)

(3) 文字處理訓練 (WordStar)

(4) Lotus 123/Symphony 操作訓練

其他電腦課程：

(1) 電腦入行職業先修課程

(2) Z－80及6502組合語言

(3) 中學會考電腦科補習

- 理論與實習並重
- 課程深入淺出，適合各階層人仕
- 每人一套電腦實習
- 結業後獲頒證書一份
- 成績優異者由本中心推薦加入電腦公司工作
- 學員可自由選擇實習時間
- 在課程期間，學員可隨時回校免費實習
- 可代公司訓練僱員
- 本中心備有中學會考電腦科參考資料
- 代購優惠服務（電腦雜誌　電腦軟件　磁碟　印字機　電腦文具　電腦書籍　磁碟機　電腦附件及各種名廠電腦）



索取章程，查詢及報名處：　專業電腦中心

九龍彌敦道486號益南華厦 3 樓 電梯 2 字　　電話：3-328248 3-318960

電腦功能人皆知
先租後買有保障

本港各行各業均趨向電腦化，但購置電腦系統除價錢昂貴外，更未必合用；再加上維修、員工訓練、軟件之配合等，每一項皆耗費大量金錢，故此很多公司對於電腦化計劃一直疑惑不決………



基於上述原因，本公司特推出電腦租用計劃，保証貴公司能以最低廉之價錢試用電腦，從而了解清楚電腦對貴公司之幫助及確信本公司服務完善，再決定是否購買如此便可避免因買錯電腦系統而招致金錢及時間之損失。

詳情如下：

1. 租用期：最短三個月，每個月只須付四佰餘元起，若三個月內決定購買，除可獲特價優待外，已付租金可自買價中減去。
2. 功用：文件處理、會計、CAD、生產及營業管理……等。
3. 保養：a）租用期內即日修理或換機；b）購買後可獲一年免費保養。
4. 其他服務：可為貴公司訓練電腦操作員工，確保貴公司能利用電腦解決辦公室工作。

超卓電腦系統有限公司
SUPREMIA SYSTEMS LTD.

九龍深水埗福華街146號
黃金商場地庫20號室
電話：3-601531, 3-617206
3-612549

Apple
迷大喜訊

精美電子公司推出兩大皇牌
Mocking Card 及 *Track Reader*

Mocking Card
$ 220

使AP II 所產生之音响效果達到現場一樣，有眞實感，尤其配合Sky Fox及Music Construction Set與街機毫不遜色。現時可配合用之Software已經非常之多，將來且有更多的軟件支持，其它非原裝線路之同類卡，有部份軟件不能操作及效果差的現象經常發生。

Speech Chip
$ 250　連碟連說明書，只須一片已可使用。

Track Reader
$ 200

磁碟讀軌器，能讀出$\frac{1}{2}$Track，可看到磁頭走動之情況，用以配合抄碟，更可達到最佳效果及可以減少抄碟的時間。

★ 另外本公司有MSX及任天堂機及遊戲機發售，全部特平。

精美電子公司　JAN MI ELECTRONIC CO.
九龍深水埗福華街黃金商場地庫48號A
Shop 48A, Basement, 146-152 Fuk Wa Street, Golden Center, Golden Building Kowloon.　Tel: 3-7289086

超級中文文字處理／檔案處理器 V3.3

最完善、最强效!!
集二大用途於一身
能處理任何形式書信文件

軟件特點：

(1) 預置定義常用句子鍵，簡化輸入。
(2) 印製硬本時，隨時調校每行字數，字距及行距，橫排或直排輸出。
(3) 可出「空格」，「，」，「、」及「：」。
(4) 每次可處理一篇多達2000字的文件，較同類軟件多出四倍。編印時，可將多篇文件連結列印，絕無中斷之弊，文件編印次數隨意。
(5) 強效的編輯功能，操作簡捷。
(6) 一張磁碟可儲存最多100份檔案文件。

中文文字處理系統所需硬件設備：

(1) 一台APPLE II或兼容電腦
(2) 可運行超級中文DOS之中文卡（如松台中文卡）
(3) 16K RAM卡
(4) 一台磁碟機
(5) 螢幕顯示器

附加硬件設備
(1) 第二台磁碟機
(2) 印字機

每套訂價港幣330元

附詳細使用說明書乙份

* Apple II is a registered trademark of Apple Computer Inc.

電腦時代讀者服務部
地址：旺角上海街438號同珍商業中心11字樓1104室
電話：3-312007

徵求拆軟件文章

徵求將GAMES 或 UTILITY 軟件拆成爲FILE或CO－PY A版本稿件。

我們歡迎讀者將市面各類有保護程式之GAMES 或 UT－ILITY 程式拆成爲FILE或COPY A版本寄來給我們。一經錄用，當致稿酬。

各位請遵照下列規則投稿：

(1) 所有稿件必須使用原稿紙打橫書寫，長短不論，但以清楚簡潔爲原則。

(2) 附上一張錄有所要解拆的原版軟件磁碟與及經成功解拆的FILE 或 COPY A 版本磁碟。所有磁碟將會於稿件刊出後退回。

(3) 附寄之磁碟請貼上LABEL 紙及清楚寫上姓名及聯絡電話。

(4) 稿件必須連同硬卡紙（避免磁碟於郵寄中途遭摺曲）。

(5) 稿件請寄往九龍中央郵政信箱71193 號，信封面請註明「拆FILE」。

適合私人及中小型公司使用

實用私人財政會計系統軟件包

只售30元（MENU另加5元）

- 用法簡易，毋須特別學習
- 處理各項支出賬項記錄
- 收入賬項記錄
- 財務預算

經銷處：電腦時代讀者服務部

地址：旺角上海街438號同珍商業中心11字樓1104室

（上海街／登打士街交界）　電話：3-312007

省時兼便利 何不利用郵購呢?!

郵購需知

所有電腦時代出版社旗下出品皆接受郵購。

郵購辦法簡易，只要塡妥下列表格及回郵地址，連同各項貨品總額加郵費及處理費，利用劃線支票或郵政滙票（請勿劃線），寄來九龍中央郵政信箱71193 號（外埠可用CPO BOX 71193, KOWLOON CENTRAL POST OFF－ICE, HONG KONG即可）。如要掛號請加付四元。

本人擬訂購下列貨物：

☐ 程式磁碟_____張（請另外註明期數）

☐ 電腦時代雜誌_____本（請另外註明期數）

☐ 微電腦技術_____本（請另外註明期數）

☐ 電腦時代精裝本_____本

☐ 電腦技術叢書_____本（請註明書名）

☐ 其他產品__________

基本郵費及處理費：

（磁碟及雜誌是分開寄出）

本地：雜誌／叢刊　收　3元
磁碟　2元
其他產品一律　3元
每多一本／一張加附 1 元

外埠：雜誌／叢刊　收　15元
磁碟　7元
每多一本雜誌加附13元
每多一張磁碟加附 5 元

外埠全部利用空郵寄出。

所有訂購處理需時三至六週。

(1) 磁碟總額　：
(2) 雜誌總額　：
(3) 技術叢刊總額：
(4) 其他產品總額：
(5) 郵費及處理費：
(6) 掛號　：

總數：

下列爲本人之回郵地址：

NAME：__________

ADDRESS：__________

爲何一開機後，磁碟機就會轉動？

APPLE開機後的秘密

一仔

大家都知道，當我們一打開APPLE 的電源後，磁碟上的DOS及HELLO 程式就會搬入記憶體內。現在就讓我們來研究一下，APPLE 怎樣做這些工作。

當我們一打開APPLE 的電源開關後，就會有一由低電位到高電位的脈波送到6502 的第四十脚，使APPLE 產生一個RESET 訊號（等於按下RESET 鍵），到記憶位置$FFFC及$FFFD 中取一重置向量位置$FA62（舊式的APPLE是$FF59） 而到此位置執行程式。所以只要我們開發$FA62的程式，便可明白AUTOSTART ROM 所增加的功能。

從程式(一)看來，發覺它只是設定輸入、輸出裝置的工作，直至到$FA85 後，把$3F3 的內容和#$A5作EOR運算，如不等於$3F4 的內容，就會跳至$FAA6的PWRUP副程式。

其實由PWRUP開始至$FAD4是一個COLD START的副程式。首先，它跳至APPLE][副程式以印出APPLE][字樣，設定第三頁的向量（使下次RESET 時可作WA—RM START），再把LOC0（$00）設定爲零（因這時的X暫存器等於零），LOC 1（$01）設定爲#C8 ，以準備測試七個槽（0 號槽不能放磁碟介面卡）。

然後設定LDY#$07 以檢查磁碟機中的ID，把LOC 1（$01）值減 1，檢查是否已測試完畢（CMP#$C0），是的話就跳至BASIC，否則檢查有否磁碟介面卡。

現在就讓我們看看它怎樣測試磁碟機：假設現在檢查N槽，首先把LOC 1（$01）值存到MSLOT（$7F8），以

程式(1)：RESET副程式

```
FA62:D8              CLD
FA63:20 84 FE        JSR SETNORM
FA66:20 2F FD        JSR INIT
FA69:20 93 FE        JSR SETVID
FA6C:20 89 FE        JSR SETKBD
FA6F:AD 58 C0        LDA SETAN0
FA72:AD 5A C0        LDA SETAN1
FA75:AD 5D C0        LDA SETAN2
FA78:AD 5F C0        LDA CLRAN3
FA7B:AD FF CF        LDA CLRROM
FA7E:2C 10 C0        BIT KBDSTRB
FA81:D8              CLD
FA82:20 3A FF        JSR BELL
FA85:AD F3 03        LDA SOFTEV+1
FA88:49 A5           EOR #$A5
FA8A:CD F4 03        CMP PWREDUP
FA8D:D0 17           BNE PWRUP
    :
FFFC:62 FA           DW RESET
```

程式(2)：COLD START副程式

```
FAA6:20 60 FB   JSR APPLE ][
FAA9:A2 05      LDX #$05
FAAB:BD FC FA   LDA $FAFC,X
FAAE:9D EF 03   STA $3EF,X
FAB1:CA         DEX
FAB2:D0 F7      BNE $FAAB
FAB4:A9 C8      LDA #$C8
FAB6:86 00      STX LOC0
FAB8:85 01      STA LOC1
FABA:A0 07      LDA #7
FABC:C6 01      DEC LOC1
FABE:A5 01      LDA LOC1
FAC0:C9 C0      CMP #$C0
FAC2:F0 D7      BEQ $FA9B
FAC4:8D F8 07   STA MSLOT
FAC7:B1 00      LDA (LOC0),Y
FAC9:D9 01 FB   CMP $FB01,Y
FACC:D0 EC      BNE $FABA
FACE:88         DEY
FACF:88         DEY
FAD0:10 F5      BPL $FAC7
FAD2:6C 00 00   JMP(LOC
```

程式(3)：第三頁向量及磁碟ID資料

```
FAFD:59 FA      > 第三頁的向量
FAFF:00 E0 45
FB02:20 FF 00   > 磁碟咭的ID資料
FB05:FF
FB06:03 FF 3C
```

程式(4)：磁碟驅動程式開始部份（第6槽）

```
C600:A2 20      LDX #$20
C602:A0 00      LDY #$00
C604:A2 03      LDX #$03
C606:86 3C      STX $3C
C608:8A         TXA
```

備以後使用；然後把CN00＋Y 值放入A暫存器中，和$FB01＋Y 的值作一比較，假如相同的話，就把 Y暫存器值減 2，繼續測試直至四個值已比較完畢（$3C，$03，$00，$20），才跳到CN 00去執行驅動磁碟程式；但只要有一個不相同的話，就會跳至下一個槽檢查。

所以爲什麼假如槽 6 和槽 5 都有磁碟機介面卡的話，槽 6 的會首先BOOT 動，因會首先檢查槽 7，然後槽 6，槽 5……。若各讀者想設計一介面卡，使得一開機就可動作的話，可以把EPROM程式的$CN01，$CN03，$CN05和$CN07 位置分別放入值#$20，00，03及3C ，那麼一開機就會執行該EPROM的程式了。■

磁碟機控制咭上P5／P6的秘密

APPLE BOOT ROM剖析

在APPLE DISK ÇONTROLL－ER卡上那枚P5 PROM，其內具有256個BYTE的機械碼。任何可以BO－OT UP的磁碟，無論是保護或是沒有保護，都必須將此段程式執行。很多人仕就是利用了此特點來進行所謂「BOOT－CODE TRACING」。「電腦時代」在前數期亦刊出過此類型的稿件，譬如解拆ELECTRONIC ARTS一文便是利用了「BOOT CODE TR－ACING」來進行的。

由於BOOT PROM在很多方面都極有用途，因此我們在這裏就把它的內容重新反滙編，藉此揭開它們工作的原理及個中的秘密。只要各位讀者細心閱讀此文，然後再細心參看以下反滙編的程式，便會知道DISK CONTROLL－ER的硬件地址（$C0n0－$C0nF）怎樣直接存取磁碟機，6和2的編碼方法，怎樣讀取ADDRESS和DATA FIELD，以增加閣下對APPLE磁碟機的概略認識。

NYBBLE 對照表

在$C600－$C620的機械碼是用來產製一個NYBBLE對照表（CON－VERSION TABLE），將之放在$36C－$3D5。這個表是用來把磁碟讀回來的編碼（或NYBBLIZED）BY－TE翻譯成6BIT的十六進制數值，其值則爲$0至$3F。但話分兩頭，這段機械碼是極之困難追踪的；如果你認識一些自命對6502程式設計相當了得的朋友，不妨把這段256個機械碼給他看看，試試他的功力是否像他說得那麼高。

這段機械碼會測試每個數值是否達到6和2編碼方法的要求：HIGH BIT SET；不能有一組兩個連續0出現；而且最少要一組連續1。當一個數值相對着一個有效的DISK BYTE被找出後，該對應翻譯的數值便會貯存在一個地址上；而這個地址值是相等於$2D6加以那個有效DISK BYTE的數值。混

```
*--------------------------------------------------------------------------*
*                         DISK CONTROLLER BOOT ROM                         *
*--------------------------------------------------------------------------*

                 .OR $C6ØØ          FOR SLOT 6
                 .TA $8ØØ           STORE IT HERE FOR NOW

*--------------------------------------------------------------------------*
*                             ZERO PAGE USAGE                              *
*--------------------------------------------------------------------------*

ØØ26- SECTOR.PTR .EQ $26    POINTS TO THE LOCATION IN MEMORY WHERE THE CURRENT SECTOR IS
                            TO BE STORED
ØØ2B- SLOT       .EQ $2B    HOLDS BOOTING SLOT NUMBER TIME 16
ØØ3C- TEMPØ      .EQ $3C    USED BY DIFFERENT ROUTINES FOR DIFFERENT THINGS
ØØ3D- SECTOR     .EQ $3D    HOLDS THE SECTOR NUMBER WE ARE LOOKING FOR
ØØ4Ø- TRK.READ   .EQ $4Ø    HOLDS THE TRACK NUMBER FOUND FROM DISKETTE
ØØ41- TRACK      .EQ $41    HOLDS THE TRACK NUMBER WE ARE LOOKING FOR

*--------------------------------------------------------------------------*
*                      DISK CONTROLLER BASE ADDRESSES                      *
*--------------------------------------------------------------------------*

CØ8Ø- MAGØ.OFF   .EQ $CØ8Ø  STEPPER MAGNET Ø OFF
CØ81- MAGØ.ON    .EQ $CØ81  STEPPER MAGNET Ø ON
CØ82- MAG1.OFF   .EQ $CØ82  STEPPER MAGNET 1 OFF
CØ83- MAG1.ON    .EQ $CØ83  STEPPER MAGNET 1 ON
CØ84- MAG2.OFF   .EQ $CØ84  STEPPER MAGNET 2 OFF
CØ85- MAG2.ON    .EQ $CØ85  STEPPER MAGNET 2 ON
CØ86- MAG3.OFF   .EQ $CØ86  STEPPER MAGNET 3 OFF
CØ87- MAG3.ON    .EQ $CØ87  STEPPER MAGNET 3 ON
CØ88- MOTOR.OFF  .EQ $CØ88  DRIVE AND MOTOR OFF
CØ89- MOTOR.ON   .EQ $CØ89  DRIVE AND MOTOR ON
CØ8A- SEL.DRV1   .EQ $CØ8A  ROUTE POWER TO DRIVE 1
CØ8B- SEL.DRV2   .EQ $CØ8B  ROUTE POWER TO DRIVE 2
CØ8C- IO1        .EQ $CØ8C  STROBE LATCH FOR I/O
CØ8D- IO2        .EQ $CØ8D  LOAD DATA LATCH
CØ8E- IO3        .EQ $CØ8E  PREPARE LATCH FOR INPUT
CØ8F- IO4        .EQ $CØ8F  PREPARE LATCH FOR OUTPUT

*--------------------------------------------------------------------------*
*                          MONITOR LOCATIONS USED                          *
*--------------------------------------------------------------------------*

FF58- IORTS      .EQ $FF58  A KNOWN RTS INSTRUCTION
FCA8- WAIT       .EQ $FCA8  WAIT FOR TIME SPECIFIED BY A-REG, X & Y UNALTERED

*--------------------------------------------------------------------------*
*                         MASS DATA STORAGE AREAS                          *
*--------------------------------------------------------------------------*

Ø1ØØ- STACK      .EQ $Ø1ØØ  65Ø2 STACK AREA STARTS HERE
Ø2D6- NIBBLE.TBL .EQ $Ø2D6  6 & 2 DECODE TABLE FROM HERE TO $3D5 (FIRST NIBBLE AT $
36C)
Ø3ØØ- BUFFER1    .EQ $Ø3ØØ  EXTRA 86 BYTES (342-256=86) OF EACH SECTOR STORED FROM
HERE TO $355
Ø8ØØ- BUFFER2    .EQ $Ø8ØØ  FIRST SECTOR OF DATA GOES TO $Ø8ØØ

*--------------------------------------------------------------------------*
*                            PR# OR IN# GOES HERE                          *
*--------------------------------------------------------------------------*
```

APPLE BOOT ROM 剖析

亂嗎？不要緊，讓我們看看以下的兩個例子：

第一個十六進制的數值能夠達到6和2要求的是＄96。在讀取翻譯表中，我們發現96＝00；所以這個00便會貯存在＄2D6＋＄96，即＄36C內。如此，＄ED亦是一個有效的DISK BYTE，而從翻譯表中，我們知道ED＝30；因此30便會貯存在＄2D6＋＄ED或＄3C3 。如果你要知道一些NYBBLE對照表的更詳細資料，就請參看原始碼列表中的附註。

堆疊的存取

ROM的下一部份是在＄C621 至＄C63A用來找出磁碟機控制咭放置在那個SLOT內，與及把磁碟機的磁頭重調至TRACK ＄0。SLOT 的編號是JSR和RTS指令至APPLE的MONITOR而獲得的。當RTS執行後，該段機械碼便會從STACK 中提取返回的地址，用它來計算出控制咭在那個SLOT內。之後，SLOT 的編號便會乘以16，然後貯存在零頁的＄2B 位置上。當SLOT編號計算出後，它的值便會傳送到X寄存器中，而在這裏便可以存取正確的硬件位置。於DRIVE 1 便開始作用，而摩打便會啓動。在未開始80個PHRASE的旅程前，磁碟機會由上至下存取PHRASE ON／PHRASE OFF ，藉此把磁頭移到磁碟的最外圍——達至TRACK 0；無論磁頭從那裏開始都會是這樣。

INITALIZE 零頁

控制咭ROM 的下一個部份是在＄C062－＄C65B，此部份極之短小。它所幹的工作就只有建立所需的零頁位置，使到TRACK ＄0，SECTOR ＄0讀進PAGE ＄8。零頁位置＄26－＄27就指着數據緩衝區（DATE BUFFER），＄3D就載着SECTOR編號，而＄41則載着TRACK編號。

讀取磁碟

在＄C65C 開始的機械碼可說是整個DISK CONTROLLER程式的戲肉。在這裏的機械碼會由DOS 3·3 的

```
C600: A2 20              LDX #$20        DOES NOTHING EXCEPT IDENTIFY DISK CONTROLLER

*-----------------------------------------------------------------------------------*
*                        SET UP A READ TRANSLATE TABLE                              *
*                             A = GENERAL USAGE                                     *
*                   X = NIBBLE TO TEST FOR VALIDITY (-$8Ø)                          *
*                 Y = WHAT NEXT VALID NIBBLE TRANSLATES INTO                        *
*-----------------------------------------------------------------------------------*

C602: AØ ØØ              LDY #$ØØ        FIRST BYTE TRANSLATES TO A $ØØ
C604: A2 Ø3              LDX #$Ø3        COULD HAVE USED ANYTHING FROM $ØØ - $16
C606: 86 3C     TST.NIBB STX TEMPØ       PUT NIBBLE HERE SO IT CAN BE MANIPULATED
C608: 8A                 TXA             PUT NIBBLE IN A SO WE CAN SHIFT IT
C609: ØA                 ASL             TEST FOR ADJACENT ONES
C60A: 24 3C              BIT TEMPØ       IF A ^ M = Ø THEN Z=1
C60C: FØ 1Ø              BEQ NXT.NIBB NO ADJACENT ONES, GET NEXT NIBBLE
C60E: Ø5 3C              ORA TEMPØ       THE ASL, ORA, EOR SEQUENCE WILL
C61Ø: 49 FF              EOR #$FF        CHANGE ALL ADJACENT Ø-BITS TO ONE 1-BIT
C612: 29 7E              AND #$7E        IGNORE BIT 7 AND Ø
C614: BØ Ø8     ADJØ.TST BCS NXT.NIBB NEVER TAKEN THE FIRST TIME THROUGH, ON SUBS
    EQUENT TIMES, THIS BRANCH IS TAKEN IF THERE STILL ARE BITS SET IN THE NIBBLE
    AND A SET BIT HAS BEEN PUT INTO THE CARRY: THEREFORE MORE THAN ONE PAIR OF ADJ
    ACENT Ø-BITS
C616: 4A                 LSR            SHIFT A BIT INTO THE CARRY, COULD HAVE USED A
    SL
C617: DØ FB              BNE ADJØ.TST TAKEN IF THERE ARE STILL SOME BITS SET (THERE
    FORE SOME ADJACENT Ø-BITS) LEFT IN NIBBLE
C619: 98                 TYA            NIBBLE PASSED TEST, A=WHAT NIBBLE TRANSLATES
    INTO
C61A: 9D 56 Ø3           STA NIBBLE.TBL+$8Ø,X STORE IT IN TRANSLATE TABLE + $8Ø
C61D: C8                 INY             NEXT NIBBLE TRANSLATES INTO Y + 1
C61E: E8        NXT.NIBB INX             GET NEXT NIBBLE TO TRY (-$8Ø)
C61F: 1Ø E5              BPL TST.NIBB IF NOT LAST NIBBLE $7F=$FF THEN KEEP GOING

*-----------------------------------------------------------------------------------*
*                           FIND WHAT SLOT WE ARE IN                                *
*-----------------------------------------------------------------------------------*

C621: 2Ø 58 FF           JSR IORTS       JSR TO A KNOWN RTS INSTRUCTION
C624: BA                 TSX             PUT STACK POINTER IN X
C625: BD ØØ Ø1           LDA STACK,X     GET MSB OF RETURN ADDRESS FROM STACK
C628: ØA                 ASL             MOVE LOWER FOUR BITS INTO UPPER FOUR BITS
C629: ØA                 ASL             IN OTHER WORDS, MULTIPLY BY 16 MOD 256
C62A: ØA                 ASL
C62B: ØA                 ASL
C62C: 85 2B              STA SLOT        SAVE SLOT NUMBER TIME 16 IN ZERO PAGE
C62E: AA                 TAX             AND PUT SLOT NUMBER * 16 IN X

*-----------------------------------------------------------------------------------*
*                                TURN ON DRIVE                                      *
*-----------------------------------------------------------------------------------*

C62F: BD 8E CØ           LDA IO3,X       SET DRIVE IN READ MODE
C632: BD 8C CØ           LDA IO1,X
C635: BD 8A CØ           LDA SEL.DRV1,X MAKE SURE DRIVE ONE!
C638: BD 89 CØ           LDA MOTOR.ON,X TURN ON MOTOR

*-----------------------------------------------------------------------------------*
*                           RECALIBRATE DRIVE HEAD                                  *
*-----------------------------------------------------------------------------------*

C63B: AØ 5Ø              LDY #8Ø         MOVE BACKWARDS OVER 8Ø PHASES (4Ø TRACKS)
C63D: BD 8Ø CØ  NXT.PHASE LDA MAGØ.OFF,X TURN CURRENT PHASE MAGNET OFF
C64Ø: 98                 TYA             CALCULATE NEXT PHASE
```

```
C641: 29 Ø3              AND #$Ø3      MOD 4
C643: ØA                 ASL           *2
C644: Ø5 2B              ORA SLOT      MERGE IN TO ZERO PAGE SLOT HOLDER
C646: AA                 TAX           PUT NEXT PHASE INTO X
C647: BD 81 CØ           LDA MAGØ.ON,X TURN ON NEXT MAGNET
C64A: A9 56              LDA #$56      WAIT DELAY OF 19.2 MILLISECONDS
C64C: 2Ø A8 FC           JSR WAIT      WAIT A WHILE
C64F: 88                 DEY           NEXT PHASE OR HALF TRACK
C65Ø: 1Ø EB              BPL NXT.PHASE KEEP GOING IF NOT DONE
*---------------------------------------------------------------------*
*          SET UP ZERO PAGE TO READ TRACK Ø, SECTOR Ø                 *
*---------------------------------------------------------------------*
C652: 85 26              STA SECTOR.PTR LSB OF POINTER = Ø
C654: 85 3D              STA SECTOR    SECTOR Ø
C656: 85 41              STA TRACK     TRACK Ø
C658: A9 Ø8              LDA /BUFFER2 MSB OF POINTER = $Ø8
C65A: 85 27              STA SECTOR.PTR+1 PUT SECTOR AT $Ø8ØØ
*---------------------------------------------------------------------*
*                 FIND EITHER THE ADDRESS HEADER                      *
*              OR DATA HEADER OF THE CURRENT SECTOR                   *
*---------------------------------------------------------------------*
C65C: 18        GET.ADDR.HDR CLC      C=Ø MEANS LOOK FOR ADDRESS HEADER
C65D: Ø8        GET.DATA.HDR PHP      SAVE CARRY
C65E: BD 8C CØ  P1       LDA IO1.X     READ DISK
C661: 1Ø FB              BPL P1        WAIT FOR FIRST BYTE
C663: 49 D5     TRY.D5   EOR #$D5      IS IT A $D5?
C665: DØ F7              BNE P1        NOPE. GET NEXT BYTE
C667: BD 8C CØ  .1       LDA IO1.X     READ DISK
C66A: 1Ø FB              BPL .1        WAIT FOR SECOND BYTE
C66C: C9 AA              CMP #$AA      IS IT A $AA?
C66E: DØ F3              BNE TRY.D5    NOPE. MAYBE IT WAS A $D5
C67Ø: EA                 NOP           WAIT 2 CYCLES
C671: BD 8C CØ  .2       LDA IO1.X     READ DISK
C674: 1Ø FB              BPL .2        WAIT FOR LAST BYTE IN EITHER ADDRESS OR DATA
  HEADER
C676: C9 96              CMP #$96      IS IT THE ADDRESS HEADER?
C678: FØ Ø9              BEQ GET.SCT.NUM YES. SEE IF RIGHT ONE
C67A: 28                 PLP           WERE WE LOOKING FOR THE DATA MARK?
C67B: 9Ø DF              BCC GET.ADDR.HDR NOPE. TRY FOR ADDRESS HEADER AGAIN
C67D: 49 AD              EOR #$AD      LOOKING FOR DATA HEADER. IS THIRD BYTE AND $A
D?
C67F: FØ 25              BEQ READ.SCT YES. GET THE DATA FROM THE SECTOR
C681: DØ D9              BNE GET.ADDR.HDR NOPE. FIND ANOTHER ADDRESS HEADER
*---------------------------------------------------------------------*
*                 READ VOLUME, TRACK AND SECTOR                       *
*---------------------------------------------------------------------*
C683: AØ Ø3     GET.SCT.NUM LDY #$Ø3   THREE 4+4 ENCODED BYTES
C685: 85 4Ø     STORE.44 STA TRK.READ STORE CURRENT DECODED BYTE HERE, FIRST VOLUME
 NUMBER WHICH WILL BE OVERWRITTEN BY THE TRACK NUMBER
C687: BD 8C CØ  .1       LDA IO1,X     READ DISK
C68A: 1Ø FB              BPL .1        WAIT FOR FIRST HALF OF BYTE
C68C: 2A                 ROL           RE-POSITION EVERY OTHER BIT
C68D: 85 3C              STA TEMPØ     SAVE IT HERE
C68F: BD 8C CØ  .2       LDA IO1,X     READ DISK
C692: 1Ø FB              BPL .2        WAIT FOR SECOND HALF OF BYTE
C694: 25 3C              AND TEMPØ     MERGE TWO HALVES TOGETHER TO DECODE BYTE
C696: 88                 DEY           DONE WITH ADDRESS FIELD (EXCEPT CHECKSUM)?
```

BOOT 1一位於＄84A（JMP（＄003E））,把RWTS讀進記憶內。一些下了保護的磁碟，如ROCKY'S BOOT，就在閱讀磁碟的時候，完全倚賴這一片DISK CONTROLLER 機械碼。

當進入了此部份後，這個副程式會首先清除CARRY FLAG以顯示一個ADDRESS HEADER 已被讀入。然而，當程式找出了D5 AA 96 的ADD－RESS HEADER後，便會分支到＄C683，而在這裏4和4編碼的VOLU－ME，TRACK，和SECTOR 數據便會讀入。TRACK和SECTOR 編號便會與 ＄3D（SECTOR）和 ＄41（TRACK）的內容比較。如果它們是正確的話，CARRY FLAG便會SET，而程式便返回到＄C65D；這次便會讀入一個DATA HEADER。如果TRA－CK和SECTOR 是不正確的話，那麼CARRY FLAG便會被清除，而該段碼便會跳到＄C65C，再次找尋指定的TRACK和SECTOR。

如果你看看＄C65C ＄C6A5 的部份，你便會發現ADDRESS EPIL－OG或CHECKSUM是從沒有被VE－RIFIED的。有些下了COPY－PR－OTECTED 的磁碟就是利用此個特點，修改了ADDRESS EPILOG 或CHECKSUM，別人利用一般的SEC－TOR EDITOR 讀取TRACK ＄0，SECTOR ＄0 的資料。可是，在TRA－CK ＄0，SECTOR ＄0 上的ADD－RESS或DATA PROLOGUES，就不能由D5 AA 96和D5 AA AD修改爲其他資料。

將數據解碼

在＄C6A6－＄C6EA 的機械碼是負責把指定SECTOR 上的編碼數據讀入適當的BUFFER 上。由於設計上是需要＄156（十進制342）個6和2的編碼BYTE 來貯存一個SECTOR 的數據，因此閱讀便會分兩次進行。

首先，＄56個BYTE會讀進＄300－＄356，其餘＄100個BYTE 則讀進＄26－＄27所指着的BUFFER 位置內（當BOOTING時，是在＄800）。注意在＄C6AA和＄C6BC處，當某個

APPLE BOOT ROM 剖析

ＢＹＴＥ的數據從磁碟讀入的時候，它便會載入Y寄存器內。這樣，從磁碟中讀入的數值便可以用作索引之用，以對應由$36C－$3D5所產製的NYBBLE對照表。就是這樣，一個有效的DISK BYTE 翻譯成一個6 BIT 的十六進數。

數據CHECKSUM

在$156個BYTE 未貯入指定的BUFFER前，每個BYTE 都會進行一個EXCLUSIVE OR操作。此個行動可說是必要的，因爲那些數據在未寫進磁碟前，那些數據都會自行和自己進行ＥＯＲ。於是這些ＥＯＲ便會產製出一個累積的CHECKSUM BYTE ，而它則會貯存在一個編碼SECTOR 的最後一個BYTE內。當SECTOR 讀入的時候，ＥＯＲ便會再次執行，如果在最後一個CHECKSUM BYTE EOR 完畢後，ACCUMULATOR所載着的數值不是零，那麽便表示有錯誤出現，而程式便會返回至$C683 ，再次試行閱讀。

如果沒有CHECKSUM ERROR 出現，那麽程式便會跳至$C6D5 ；在這裏，程式便會把貯存在兩個BUFFER中的6和2編碼數據，還原爲原本的非編碼形式。其間的過程牽涉了很多的BIT－SHIFTING，因爲每個ＢＹＴＥ的其中兩個BIT在寫入磁碟前，被隔離到其他的ＢＹＴＥ處。然而，這兩個BIT是來自貯存在$300－$355 的數據中。

在主BUFFER（$800－$8FF）的數據被還原至原本的模式後，那些機械碼便會跳至$C6EB。在這裏，程式便會查看一下，看看要否繼續讀取其他的SECTOR 。讀取SECTOR 的數目是貯存在$800內。通常，這個BYTE會載着一個1，但它的最大值則可以達至$F 。筆者只發現一些少數的磁碟在它們STAGE 0 BOOT UP 時，載入多一個SECTOR 以上。一旦當指定數目的SECTOR 讀入後，JMP $801便會執行，開始下一個STAGE 的BO－OT UP 。

當跳至$801的指令執行後，磁碟便可以任由驅策；然而大部份磁碟都會利用由$C65C 來在TRACK $0讀入數個SECTOR，正常的DOS 3・3 磁碟則會在TRACK $0，SECTOR $0－$9 讀入RWTS 進記憶內。 ■

```
C697: D0 EC                  BNE STORE.44 NOPE, GET NEXT BYTE

*-------------------------------------------------------------------------*
*                    SEE IF THIS IS THE SECTOR WE WANT                    *
*-------------------------------------------------------------------------*

C699: 28                     PLP          FIX STACK BECAUSE OF THE PHP AT $C65D
C69A: C5 3D                  CMP SECTOR   A-REG HAS SECTOR READ, IS IT THE RIGHT ONE?
C69C: D0 BE                  BNE GET.ADDR.HDR NOPE, LOOK FOR NEXT ADDRESS FIELD
C69E: A5 40                  LDA TRK.READ GET TRACK NUMBER READ
C6A0: C5 41                  CMP TRACK    IS IT THE ONE WE WANT?
C6A2: D0 B8                  BNE GET.ADDR.HDR NOPE, LOOK FOR NEXT ADDRESS FIELD
C6A4: B0 B7                  BCS GET.DATA.HDR ..ALWAYS TAKEN, LOOK FOR DATA FIELD HEADE
 R

*-------------------------------------------------------------------------*
*                    READ SECTOR AND STORE IT IN MEMORY                   *
*-------------------------------------------------------------------------*

C6A6: A0 56       READ.SCT LDY #86        READ FIRST EIGHTY SIX BYTES
C6A8: 84 3C       NXT.BUF1 STY TEMP0      SAVE NUMBER OF BYTES OF BUFFER1 LEFT TO READ
C6AA: BC 8C C0    .1       LDY IO1,X      READ DISK
C6AD: 10 FB                BPL .1         WAIT FOR 6 & 2 ENCODED NIBBLE OF DATA
C6AF: 59 D6 02             EOR NIBBLE.TBL,Y DECODE THE BYTE (A=0 FIRST TIME THROUGH)
C6B2: A4 3C                LDY TEMP0      GET NUMBER OF BYTES LEFT
C6B4: 88                   DEY            DONE WITH BUFFER1?
C6B5: 99 00 03             STA BUFFER1,Y SAVE 6 BIT NIBBLE
C6B8: D0 EE                BNE NXT.BUF1 NOT DONE WITH FIRST 86 BYTES, CONTINUE

*-------------------------------------------------------------------------*
*                    READ REMAINING DATA FIELD BYTES                      *
*-------------------------------------------------------------------------*

C6BA: 84 3C       NXT.BUF2 STY TEMP0      SAVE THE CURRENT BYTE NUMBER (FIRST=0)
C6BC: BC 8C C0    .1       LDY IO1,X      READ DISK
C6BF: 10 FB                BPL .1         WAIT FOR BYTE
C6C1: 59 D6 02             EOR NIBBLE.TBL,Y TRANSLATE NIBBLE
C6C4: A4 3C                LDY TEMP0      GET CURRENT BYTE INDEX (NUMBER)
C6C6: 91 26                STA (SECTOR.PTR),Y SAVE BYTE IN BIG BUFFER
C6C8: C8                   INY            NEXT BYTE INDEX
C6C9: D0 EF                BNE NXT.BUF2 KEEP READING BYTES UNTIL ALL 256 HAVE BEEN RE
 AD
C6CB: BC 8C C0    .2       LDY IO1,X      READ DISK
C6CE: 10 FB                BPL .2         WAIT FOR DATA FIELD CHECKSUM
C6D0: 59 D6 02             EOR NIBBLE.TBL,Y TRANSLATE THE BYTE
C6D3: D0 87       GOTO.NXT.SECTOR BNE GET.ADDR.HDR IF CHECKSUM NOT RIGHT THEN START A
 LL OVER

*-------------------------------------------------------------------------*
*                    DECODE THE 6 & 2 ENCODED SECTOR                      *
*-------------------------------------------------------------------------*

C6D5: A0 00                LDY #$00       START WITH BYTE 0 OF THE BUFFER
C6D7: A2 56       STRT.BUF1 LDX #86       86 BYTES IN BUFFER1
C6D9: CA          NXT.62.DEC DEX          GET NEXT BUFFER1 INDEX
C6DA: 30 FB                BMI STRT.BUF1 IF = $FF THEN START BACK AT END OF BUFFER1
C6DC: B1 26                LDA (SECTOR.PTR),Y GET BYTE MISSING LOWER TWO BITS
C6DE: 5E 00 03             LSR BUFFER1,X GET BIT FROM BUFFER1
C6E1: 2A                   ROL            PUT INTO BUFFER2 BYTE
C6E2: 5E 00 03             LSR BUFFER1,X GET THE OTHER BIT FROM BUFFER1
C6E5: 2A                   ROL            PUT INTO BUFFER2 BYTE
C6E6: 91 26                STA (SECTOR.PTR),Y BYTE IS DECODED, STORE IT
C6E8: C8                   INY            GET NEXT BUFFER2 INDEX
C6E9: D0 EE                BNE NXT.62.DEC NOT DONE WITH WHOLE PAGE, DO NEXT BYTE

*-------------------------------------------------------------------------*
*                    TEST FOR ANOTHER SECTOR                              *
*-------------------------------------------------------------------------*

C6EB: E6 27                INC SECTOR.PTR+1 NEXT BUFFER MSB
C6ED: E6 3D                INC SECTOR     NEXT SECTOR NUMBER
C6EF: A5 3D                LDA SECTOR     GET SECTOR NUMBER
C6F1: CD 00 08             CMP BUFFER2    HAVE WE FINISHED READING?
C6F4: A6 2B                LDX SLOT       GET SLOT # TIME 16
C6F6: 90 DB                BCC GOTO.NXT.SECTOR LESS THAN SECTOR SPECIFIED, READ NEXT
 ONE
C6F8: 4C 01 08             JMP BUFFER2+1 ENOUGH SECTORS, CONTINUE WITH BOOT
C6FB: 00 00 00
C6FE: 00 00                .HS 0000000000 UNUSED BYTES
```

PRINTSHOP助手

自製私人名片及信紙

黃偉江

本期筆者爲大家介紹一個非常實用的程式，名爲PRINT－SHOP ASSISTANT 。藉此可以輕易地自行印製一些私人名片（見圖一），私人信紙（圖二）及能把一個PRINT－SHOP GRAPHIC 反白（白變黑，黑變白），相信各位都會感到興趣。

此程式有兩個特點，第一是能夠應用PRINTSHOP 的GRAPHIC於印製名片及信紙。例如附圖的電腦時代標誌是筆者利用PRINTSHOP的GRAPHIC EDITOR 來製作的，同樣各位亦可以自行製作一些自己的標誌供本程式印出。

另一方面本程式雖然控制了印字機印出文字及圖案，但整個程式沒有一行是PR#1，換言之是以機械語言來驅動印字機；優點是保持了畫面的完整性。

硬件要求

程式之最基本要求是一具與EPSON印字機指令相容的印字機及插在SLOT 1 的EPSON或GRAPPLER介面。筆者編寫此程式之時是以KX－P1091／EN1091 爲準，而附圖亦是由1091印出，取其有LETTER QUALITY字體。但當筆者使用CP－80 印出之時，發覺CP－80 和1091的每行點數不同，要把程式略作修改才得出滿意的效果。其他的印字機筆者並沒有試過，可能尺寸會有所差異，又或者根本不能出GRAPHIC。無論如何，各位在使用本程式之時遇到有什麼問題，可來信電腦時代轉筆者收。

名片／信紙格式

各位可從附圖看到，無論是名片（NAME CARD）或信紙，都是一個PRINTSHOP GRAPHIC 及六行文字。這六行文字頭三行是大字（LINE 1－3），每行十五字；及三行小字（LINE 4－6），每行三十字。大字小字都可選擇NORMAL或ITALIC 字體，如讀者擁有1091更可選擇LE－TTER QUALITY 字體，使印出的字形更爲美觀。

執行／鍵入程式

讀者要特別留意，收錄在即期程式磁碟及列表一的PRI－NTSHOP ASSISTANT主程式是CP－80 印字機的版本。如讀者的印字機是1091，要把行70改作

70　LG＝406：SP＝8：CL＝12：XL＝960：NL＝25

才能得出預期的結果。

另外本文還有一個機械碼列表（列表二），每行冒號後的數值是CHECK SUM， 如使用第十五期的機械碼輸入編輯器可以一面打一面核對，避免了錯誤。又或者可在MONI－TOR 直接鍵入（不必打入CHECK SUM），之後便以



圖(1)　利用本文程式印製的私人名片

BSAVE PSA·OBJ，A＄300，L＄AC存入磁碟。

把兩個列表PRINTSHOP ASSISTANT及PSA·OBJ 都存好在磁碟裏便可試RUN 了。本程式十分簡單，一看便明，如輸入打錯會有獨特聲響提示。首先是功能的選擇，跟着便要輸入一個使用PRINTSHOP GRAPHIC 的名字。這裏所指PRINTSHOP GRAPHIC 是DOS 3·3格式存儲的一種，例如在PRINTSHOP GRAPHIC LIBR－ARY（BRODERBUND公司出品，原裝兩面碟）內便有很多GRAPHIC，而不是在PRINTSHOP 原碟那些GRA－PHIC。不過如要使用PRINTSHOP原碟的GRAPHIC，可先經SECTOR EDITOR SAVE 在 DOS 3·3 格式的磁碟上才取用。

提起了PRINTSHOP GRAPHIC LIBRARY ，筆者發現原裝的一隻（2面碟）GRAPHIC非常精緻，非以前的一隻"假嘢"（1面碟）可比。如以本程式配合原裝的PRI－NTSHOP GRAPHIC LIBRARY定必生色不少。

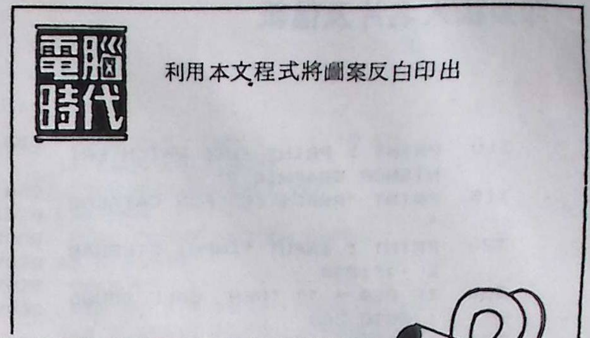
輸入了GRAPHIC的名字後，便輪到文字了。剛才提過大字有三行（LINE 1－3），每行十五字；小字三行（LI－NE 4－6），每行三十字，過長之輸入會自動縮短。而程式會自動校中間位（AUTO－CENTERING） 使左右對稱，令文字更爲美觀。最後是輸入印製的數量，可打入０－99之值，印製之時一個計數器會記錄下數量。

結語

本文旨在介紹和解釋如何使用PRINTSHOP ASSIST－ANT ，至於程式原理可參閱同期另文"活用印字機"。

有數點要補充的是在印製中切忌以CTRL－C 來中止程式，而是要按＜ＥＳＣ＞。另外在印製前最好先關掉印字機再開，以免印出一堆垃圾。行70中變數ＮＬ的值代表了信紙印出有多少條綫，讀者可隨便更改。■

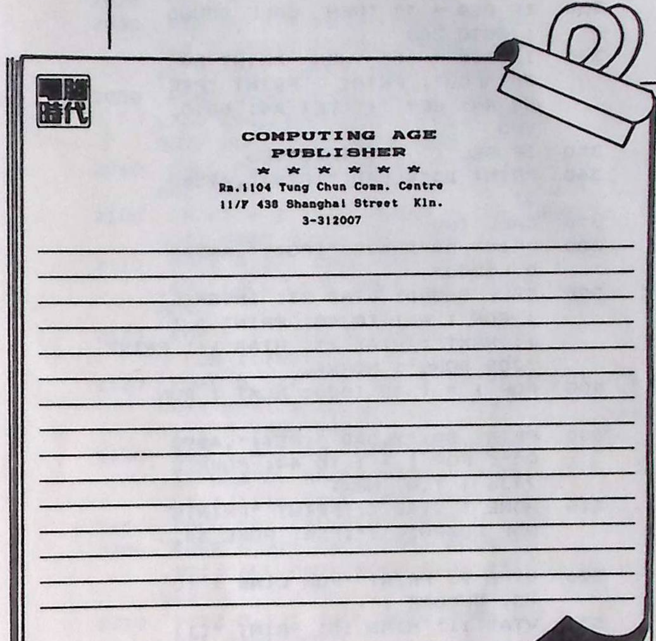
印製私人名片及信紙



利用本文程式將圖案反白印出

表一

```
****************
*   PSA.OBJ    *
* A$300,L$AC   *
****************
0300-A9 0A 85 00 A5 00 2C C1 :CA
0308-C1 30 FB 8D 90 C0 60 00 :29
0310-00 00 00 00 00 00 00 A9 :A9
0318-00 85 FB A2 00 A0 00 B1 :73
0320-06 9D 0F 03 E8 98 18 69 :B6
0328-0B A8 E0 08 D0 F1 A2 00 :FE
0330-A9 00 2E 0F 03 90 02 09 :84
0338-80 2E 10 03 90 02 09 40 :9C
0340-2E 11 03 90 02 09 20 2E :2B
0348-12 03 90 02 09 10 2E 13 :01
0350-03 90 02 09 08 2E 14 03 :EB
0358-90 02 09 04 2E 15 03 90 :75
0360-02 09 02 2E 16 03 90 02 :E6
0368-09 01 20 06 03 E8 E0 08 :03
0370-D0 BE E6 06 D0 02 E6 07 :39
0378-E6 FB A5 FB C9 0B D0 9B :C0
0380-60 A0 D0 A9 08 20 A8 FC :45
0388-AD 30 C0 88 D0 F5 60 A9 :F3
0390-58 85 07 A9 00 85 06 A8 :C0
0398-A9 FF 38 F1 06 91 06 E6 :54
03A0-06 D0 F5 E6 07 A5 07 C9 :2D
03A8-61 D0 ED 60 20 21 FF 22 :E0
```



圖(2) 利用本文程式印製的私人信紙

表二

```
0   REM **************************
1   REM *                        *
2   REM * PRINTSHOP---ASSISTANT *
3   REM *                        *
4   REM * (C)1985 BY COMPUHOUSE *
5   REM *                        *
6   REM * AUTHOR: WONG WAI KONG *
7   REM *                        *
8   REM **************************
9   REM
10   CLEAR :D$ =  CHR$ (4)
20  FT$(1) = "NORMAL":FT$(2) = "IT
     ALIC"
30  RM$(1) = "NOW TYPE YOUR MESSAG
     E, BUT DO NOT EXCEED LIMIT B
     ELOW:"
40  CH$(1) = "NORMAL":CH$(2) = "IT
     ALIC":CH$(3) = "NLQ (KX/EP 1
     091N ONLY)"
50  SEL$(1) = "PRINT  NAME CARD":S
     EL$(2) = "PRINT LETTER PAPER
     "
60  LF = 768:OUT = 772:PG = 791:SO
     UND = 897:INV = 911
70  LG = 538:SP = 7:CL = 16:XL = 1
     920:NL = 25
90   IF  PEEK (768) <  > 169 THEN
       PRINT  CHR$ (13); CHR$ (4)"
     BLOAD PSA.OBJ,A$300"
100  TEXT : HOME : INVERSE
110  FOR Y = 1 TO 6: VTAB Y: FOR
     X = 1 TO 40: PRINT " ";: NEXT
     : NEXT
115  HTAB 1: VTAB 1: PRINT ":": HTAB
     40: VTAB 1: PRINT ":": HTAB
     1: VTAB 6: PRINT ":": HTAB 4
     0: VTAB 6: PRINT ":"
120  VTAB 2: HTAB 9: PRINT "--PRI
     NTSHOP  ASSISTANT--"
130  VTAB 4: HTAB 4: PRINT "COPYR
     IGHT (C)1985 BY COMPUTING AG
     E"
135  VTAB 5: HTAB 10: PRINT "AUTH
     OR:  WONG WAI KONG"
150  NORMAL : POKE 34,6
200  VTAB 9: HTAB 10: PRINT "[1]
     ";SEL$(1)
210  VTAB 11: HTAB 10: PRINT "[2]
      ";SEL$(2)
220  VTAB 13: HTAB 10: PRINT "[3]
      INVERT A PRINTSHOP GRAPHIC"
230  VTAB 14: HTAB 10: PRINT "
      AND SAVE IT BACK."
235  VTAB 16: HTAB 10: PRINT "[4]
     QUIT"
240  VTAB 19: PRINT "CHOOSE ONE F
     UNCTION ->";: GET A$: PRINT
     A$
250 SEL =  VAL (A$): IF SEL > 4 OR
     SEL < 1 THEN  CALL SOUND: GOTO
     240
260  IF SEL = 4 THEN 9000
300  HOME
```

```
310  PRINT : PRINT "USE WHICH PRI
     NTSHOP GRAPHIC ?"
315  PRINT "PRESS <C> FOR CATALOG
     "
320  PRINT : INPUT "INPUT FILENAM
     E ->";PS$
330  IF PS$ = "" THEN  CALL SOUND
     : GOTO 300
340  IF PS$ = "C" THEN  PRINT D$"
     CATALOG": PRINT : PRINT "PRE
     SS ANY KEY ";: GET A$: GOTO
     300
350  IF SEL <  > 3 THEN 500
360  PRINT D$"BLOAD ";PS$;",A$580
     0"
370  CALL INV
380  PRINT D$"BSAVE ";PS$;",A$580
     0,L$240"
390  CALL SOUND: VTAB 23: INVERSE
     : FOR I = 1 TO 40: PRINT " "
     ;: NEXT : VTAB 23: HTAB 16: PRINT
     "JOB DONE": NORMAL
400  FOR I = 1 TO 1000: NEXT : RUN

500  PRINT D$;"BLOAD ";PS$;",A$90
     00": FOR I = 1 TO 44: POKE 3
     7436 + I,0: NEXT
510  HOME : VTAB 7: PRINT "PRINTS
     HOP GRAPHIC :";PS$: POKE 34,
     7
520  VTAB 9: PRINT "FOR LINE 1 TO
     3, CHOOSE :"
530  VTAB 11: HTAB 15: PRINT "[1]
     ";FT$(1)
540  PRINT : HTAB 15: PRINT "[2]
     ";FT$(2)
550  VTAB 15: PRINT "ENTER A NUMB
     ER ->";: GET A$: PRINT A$
560 FT =  VAL (A$): IF FT <  > 2 AND
     FT <  > 1 THEN  CALL SOUND: GOTO
     550
570  HOME : VTAB 8: HTAB 10: PRINT
     "LINE 1-3 :";FT$(FT): POKE 3
     4,8
580  VTAB 10: PRINT RM$(1)
590  FOR L = 1 TO 3: VTAB 13 + L *
     2: HTAB 10: FOR I = 1 TO 15:
      PRINT "'";: NEXT
595  VTAB 12 + L * 2: HTAB 1: PRINT
     "LINE ";L;" ->";: INPUT "";L
     INE$(L):LINE$(L) =  LEFT$ (L
     INE$(L),15): NEXT
600  HOME : VTAB 10: PRINT "FOR L
     INE 4 TO 6, CHOOSE :"
610  FOR I = 1 TO 3: VTAB 10 + I *
     2: HTAB 10: PRINT "[";I;"] "
     ;CH$(I): NEXT
620  VTAB 18: PRINT "CHOOSE A NUM
     BER ->";: GET A$: PRINT A$
630 CH =  VAL (A$): IF CH > 3 OR
     CH < 1 THEN  CALL SOUND: GOTO
     620
640  HOME : VTAB 9: HTAB 10: PRINT
     "LINE 4-6 :";CH$(CH): POKE 3
     4,9
645  VTAB 11: PRINT RM$(1)
650  FOR L = 4 TO 6: VTAB 14 + (L
     - 3) * 2: HTAB 10: FOR I =
     1 TO 30: PRINT "'";: NEXT
655  VTAB 13 + (L - 3) * 2: HTAB
     1: PRINT "LINE ";L;" ->";: INPUT
     "";LINE$(L):LINE$(L) =  LEFT$
     (LINE$(L),30): NEXT
660  VTAB 21: INPUT "AND FINALLY
     ENTER NUMBER OF COPIES TO BE
      PRINTED (1-99) ->";NO
670  IF NO > 99 OR NO < 1 THEN  CALL
     SOUND: GOTO 660
680  HOME : VTAB 10: HTAB 5: PRINT
     "NO. OF COPIES :";NO: POKE 3
     4,10
690 Z = (40 -  LEN (SEL$(SEL))) /
     2 - 1: INVERSE : VTAB 11: PRINT
     ":"; SPC( Z);SEL$(SEL); SPC(
     Z);":": POKE 34,11
700  REM  START PRINTING
710  POKE 0,27: CALL OUT
720  POKE 0,65: CALL OUT
730  POKE 0,SP: CALL OUT: REM  LF
      SPACE
740  REM
800  REM  RESET TO NORMAL
810  POKE 0,18: CALL OUT
820  POKE 0,27: CALL OUT: POKE 0,
     53: CALL OUT
830  POKE 0,27: CALL OUT: POKE 0,
     72: CALL OUT
900  REM
910  REM  PRINT COUNTER
920  REM
930  FOR PC = 1 TO NO
940  GOSUB 7000: ON SEL GOSUB 100
     0,2000
950 K =  PEEK ( - 16384): POKE  -
     16368,0: IF K = 155 THEN  RUN

960  NEXT PC: RUN
1000  REM  PRINT NAME CARD
1010  CALL LF
1020 LE = LG:DIR = 1: GOSUB 1500:
      CALL LF
1025  GOSUB 1600: CALL LF
1030 AD = 36864
1050  FOR PS = 1 TO 7: POKE 7, INT
      (AD / 256): POKE 6,((AD / 25
      6) -  INT (AD / 256)) * 256
1070  POKE 0,27: CALL OUT: POKE 0
      ,76: CALL OUT: POKE 0,((LG /
      256) -  INT (LG / 256)) * 25
      6: CALL OUT: POKE 0, INT (LG
      / 256): CALL OUT
1080  POKE 0,255: CALL OUT: POKE
      0,0: FOR I = 1 TO 24: CALL O
      UT: NEXT : CALL PG: POKE 0,0
      : FOR R = 1 TO LG - 114: CALL
      OUT: NEXT : POKE 0,255: CALL
      OUT: CALL LF
1090 AD = AD + 88: NEXT
1110  IF FT = 2 THEN  POKE 0,27: CALL
      OUT: POKE 0,52: CALL OUT
1120  POKE 0,27: CALL OUT: POKE 0
      ,87: CALL OUT: POKE 0,1: CALL
      OUT
1130  FOR L = 1 TO 3:LS = (LG - C
      L * 2 *  LEN (LINE$(L))) / 2
      : GOSUB 1700
1135  IF  LEN (LINE$(L)) = 0 THEN
      1150
```

印製私人名片及信紙

```
1140  FOR C = 1 TO  LEN (LINE$(L)
     ): POKE 0, ASC ( MID$ (LINE$
     (L),C,1)): CALL OUT: NEXT
1150 RS = LS: GOSUB 1800: CALL LF
1160  GOSUB 1600: CALL LF: NEXT
1200  POKE 0,27: CALL OUT: POKE 0
     ,87: CALL OUT: POKE 0,0: CALL
     OUT
1210  POKE 0,27: CALL OUT: POKE 0
     ,53: CALL OUT
1220  IF CH = 2 THEN  POKE 0,27: CALL
     OUT: POKE 0,52: CALL OUT
1230  IF CH = 3 THEN  POKE 0,27: CALL
     OUT: POKE 0,110: CALL OUT
1240  FOR L = 4 TO 6:LS = (LG - C
     L *  LEN (LINE$(L))) / 2: GOSUB
     1700
1245  IF  LEN (LINE$(L)) = 0 THEN
     1260
1250  FOR C = 1 TO  LEN (LINE$(L)
     ): POKE 0, ASC ( MID$ (LINE$
     (L),C,1)): CALL OUT: NEXT
1260 RS = LS: GOSUB 1800: CALL LF
1270  GOSUB 1600: CALL LF: NEXT
1300 DIR = 128: GOSUB 1500: CALL
     LF
1400  RETURN
1500  REM  PRINT A LINE
1510  POKE 0,27: CALL OUT
1520  POKE 0,76: CALL OUT
1530  POKE 0,((LE / 256) -  INT (
     LE / 256)) * 256: CALL OUT
1540  POKE 0, INT (LE / 256): CALL
     OUT
1550  POKE 0,DIR: FOR I = 1 TO LE
     : CALL OUT: NEXT
1560  RETURN
1600  REM  PRINT SEP
1610  POKE 0,27: CALL OUT
1620  POKE 0,76: CALL OUT
1630  POKE 0,((LE / 256) -  INT (
     LE / 256)) * 256: CALL OUT
1640  POKE 0, INT (LE / 256): CALL
     OUT
1650  POKE 0,255: CALL OUT
1660  POKE 0,0: FOR I = 2 TO LE -
     1: CALL OUT: NEXT
1670  POKE 0,255: CALL OUT
1680  RETURN
1700  REM  PRINT LEFT
1710  POKE 0,27: CALL OUT
1720  POKE 0,76: CALL OUT
1730  POKE 0,((LS / 256) -  INT (
     LS / 256)) * 256: CALL OUT
1740  POKE 0, INT (LS / 256): CALL
     OUT
1750  POKE 0,255: CALL OUT
1760  POKE 0,0: FOR I = 2 TO LS: CALL
     OUT: NEXT
1770  RETURN
1800  REM  PRINT RIGHT
1810  POKE 0,27: CALL OUT
1820  POKE 0,76: CALL OUT
1830  POKE 0,((RS / 256) -  INT (
     RS / 256)) * 256: CALL OUT
1840  POKE 0, INT (RS / 256): CALL
     OUT
1850  POKE 0,0: FOR I = 1 TO RS -
     1: CALL OUT: NEXT
1860  POKE 0,255: CALL OUT
1870  RETURN
2000  REM  PRINT LETTER PAPER
2020 AD = 36864
2025  CALL LF
2030  FOR PS = 1 TO 7: POKE 7, INT
     (AD / 256): POKE 6,((AD / 25
     6) -  INT (AD / 256)) * 256
2050  POKE 0,27: CALL OUT: POKE 0
     ,76: CALL OUT: POKE 0,88: CALL
     OUT: POKE 0,0: CALL OUT
2060  CALL PG: CALL LF:AD = AD +
     88: NEXT
2100  IF FT = 2 THEN  POKE 0,27: CALL
     OUT: POKE 0,52: CALL OUT
2110  POKE 0,27: CALL OUT: POKE 0
     ,87: CALL OUT: POKE 0,1: CALL
     OUT
2120  FOR L = 1 TO 3:S = 20 -  INT
     ( LEN (LINE$(L)) / 2)
2130  POKE 0,32: FOR I = 1 TO S: CALL
     OUT: NEXT : IF  LEN (LINE$(L
     )) = 0 THEN 2150
2140  FOR I = 1 TO  LEN (LINE$(L)
     ): POKE 0, ASC ( MID$ (LINE$
     (L),I,1)): CALL OUT: NEXT
2150  CALL LF: CALL LF: NEXT
2200  POKE 0,27: CALL OUT: POKE 0
     ,87: CALL OUT: POKE 0,0: CALL
     OUT
2210  POKE 0,27: CALL OUT: POKE 0
     ,53: CALL OUT
2220  IF CH = 2 THEN  POKE 0,27: CALL
     OUT: POKE 0,52: CALL OUT
2230  IF CH = 3 THEN  POKE 0,27: CALL
     OUT: POKE 0,110: CALL OUT
2250  FOR L = 4 TO 6:S = 40 -  INT
     ( LEN (LINE$(L)) / 2)
2260  POKE 0,32: FOR I = 1 TO S: CALL
     OUT: NEXT : IF  LEN (LINE$(L
     )) = 0 THEN 2280
2270  FOR I = 1 TO  LEN (LINE$(L)
     ): POKE 0, ASC ( MID$ (LINE$
     (L),I,1)): CALL OUT: NEXT
2280  CALL LF: CALL LF: NEXT
2290  CALL LF
2300 LE = XL:DIR = 1
2310  FOR L = 1 TO NL
2320  GOSUB 1500: CALL LF: CALL L
     F: CALL LF
2330  NEXT
2400  RETURN
2500  REM
7000  CALL SOUND
7100  VTAB 23: INVERSE : PRINT ":
        P R I N T   C O U N T E
     R ->"; SPC( PC < 10);PC;"
     ": NORMAL : POKE 1911,58
7200  RETURN
9000  TEXT : HOME : FOR I = 37 TO
     1 STEP  - 1: SPEED= 175: VTAB
     24: HTAB I: PRINT "*";: FLASH
     : PRINT " ";: NORMAL : CALL
      - 868: NEXT : SPEED= 255: VTAB
     23
9010  FOR I = 1 TO 800: NEXT : CALL
     SOUND: POKE 2000,221: END
```

■

不同機 BIT IMAGE 印刷功能差異分析

活用印字機

◀黃偉江▶

在本文的篇幅裏，筆者打算對印字機的使用作較深入的探討，特別着眼於印字機的點映像（BIT－IMAGE） 印刷功能。本文所附的程式適用於一般的DOT－MATRIX 印字機及界面，各位可將其運用於自己的程式之中。要執行原始碼列表的程式，可先打入CALL－151 進入監視器；跟着照着列表左面的機械碼鍵入，最後當然是把其存在磁碟中。

機械語言與印字機

大家都知道如果要啓動印字機，便要打入PR＃1 。但是有沒有想過使用機械語言來驅動印字機？使用機械語言的優點十分明顯，第一當然是其快速的執行，這點在作大量印刷如BIT IMAGE印刷便十分重要了。另一方面如使用機械語言把字符送至印字機，則可儘量保存了畫面的完整性。

列表一的副程式作用就是把一BYTE資料送至印字機，要把其存在磁碟，可於鍵入後打進BSAVE BYTEOUT，A＄300，L＄0B 。

這個程式是最基本的部份，如欲在BASIC 中呼叫此副程式，可使用：

POKE 0，BYTE：CALL 768

BYTE是一個變數，存了要送至印字機字符的ASCII碼（請參閱印字機的手册） 。

一個程式例子

列表二是一個文字幕畫面傾印程式（SCREEN DUMP），十分實用，使用時可在適當的BASIC 敍述加進CALL 768便可。各位鍵入後以BSAVE DUMP 40，A＄300，L＄28存入磁碟。

大家可從列表中看到，程式的結構非常簡單；主要分爲兩個廻路，分別爲LOOPX及LOOPY。LOOPY 負責把一行（40個字）印出，而取得所印刷的字符是透過JSR BAS－CALC（＄FBC1） 得出每一行的開始地址。有一點要留意的是從記憶中取得的字符如是NORMAL 者，BIT 7一定是ON（大於127），但送至印字機之時則要使BIT 7 OFF，故筆者在行19加了一句AND ＃＄7F 以解決問題。

PRINTSHOP GRAPHIC

以下的部份將論及如何把一個PRINTSHOP GRAPH－IC 印出，首先談談其記錄方式。

每個PRINTSHOP GRAPHIC 記錄在磁碟上的長度爲576 BYTES （4個磁區）。但構成PRINTSHOP GRA－PHIC 爲88×52點，以一BYTE 8點來計，只須572 BYTES ，換言之最後的4 BYTES 是沒有用的。

它的地址由＄5800開始，而每行連續下去的，即第一行的地址是＄5800，第二行是＄580A ，餘此類推。

印出 PRINTSHOP GRAPHIC

一般的DOT MATRIX印字機的印字頭是有九支針的，但在BIT IMAGE 印刷狀態下只有八支針受軟件控制。要啓動BIT IMAGE 狀態，要把下列的控制碼送至印字機：

1. 單密度，控制碼ESC ＋K
2. 雙密度，控制碼ESC ＋L
3. 某些印字機如1091等還能有四倍密度，控制碼爲 ESC ＋Z

跟着便要告知要印多少點，因爲要印的點數大於255，所以便需要兩位元，先把低位元送至印字機，再送高位元。最後當然是把那些資料逐一送至印字機。這些資料是以印字頭來看，最上的一支針是BIT 7，最下的一支針是BIT 0，要該針打印則在相應的BIT 塡上0。這樣做了之後再把一個LINEFEED 碼（CTRL－J)送至印字機就能印出一行的BIT IMAGE，要印PRINTSHOP GRAPHIC 便要做（52除8）7 次之多。

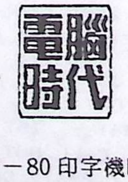
列表三是筆者所編寫的一個BASIC 程式用以印出一個指定的PRINTSHOP GRAPHIC ，而列表四則是同期「PRINTSHOP 助手」一文中一段機械碼程式的原始碼（SO－URCE CODE）。這裏刊出是爲了解釋如何能印出PRIN－TSHOP GRAPHIC，又列表三之BASIC 程式是需要一個名爲PSA•OBJ 的檔案（可從PRINTSHOP 助手一文中找到） 。

從列表四可以看到，行50之APSLINE 就是負責把11×8 BYTE 的資料翻譯爲88個資料，這是透過不斷的ASL 而得出的。筆者不打算再加解釋了，請自行參考原始碼上的一些註解。

印字機共通性

市面上出售與EPSON指令相容的印字機，它們之間其實是有所分別的，這點在BIT IMAGE印刷之時特別明顯。圖A及B是兩部印字機所印出PRINTSHOP GRAPHIC 的比較： （同樣是88×52點）

圖A



以CP－80 印字機印出

圖B



以1091印字機印出

這是因爲不同廠家生產的印字機的點密度(DPI－DOTS PER INCH)是未必相同的，故在編寫有關的程式時就要留意了。就算是印字機內置的字體，CP－80 字體每個佔 8 點，而1091的字體佔 6 點，可見是有所不同的。

另外一方面，作BIT IMAGE印刷時通常要另行設定換行送紙量，指令爲：

ESC＋A＋7　CP－80 印字機
ESC＋A＋8　1091印字機

如換行送紙量不符則會出現走針現像或是圖像擠在一起，故要小心。以上是筆者使用印字機時得出來的一些經驗，希望對各位有所帮助。

在即期程式磁碟共有六個檔案是屬於本文的。LISTING 1，2及4是LISA的原始碼，僅作參考之用。LISTING 3 RUN 了之後能把一個指定的PRINTSHOP GRAPHIC 印出印字機。DUMP 40 在BRUN之時會把文字畫面傾印出來，而BYTEOUT 只是一個副程式，BRUN 這個副程式是看不到什麽的。■

列表一

```
0800          *
0800          * THIS ROUTINE SEND A BYTE TO PRINTER
0800          * USE FROM BASIC :
0800          * POKE 0,BYTE : CALL 768
0800          *
0300                         ORG $300
0300                         OBJ $300
0300          *
0300 A5 00    BYTEOUT        LDA $00         ;GET A BYTE
0302 2C C1 C1 TEST           BIT $C1C1       ;TEST IF PRINTER IS BUSY
0305 30 FB                   BMI TEST        ;YES, THEN WAIT
0307 8D 90 C0                STA $C090       ;NO, THEN SEND
030A 60                      RTS
030B                         END

***** END OF ASSEMBLY
```

列表二

```
0800          *
0800          * USE THIS ROUTINE TO
0800          * DUMP 40 COLUMN SCREEN
0800          * USE FROM BASIC :
0800          * CALL 768
0800          *
FBC1          BASCALC        EQU $FBC1
0028          BASL           EPZ $28
0029          BASH           EPZ $29
0800          *
0300                         ORG $300
0300                         OBJ $300
0300          *
0300 A2 00    DUMP40         LDX #$00        ;INIT VERT. POS
0302 8A       LOOPX          TXA
0303 20 C1 FB                JSR BASCALC     ;CAL. BASE ADDRESS
0306 A0 00                   LDY #$00        ;INIT HORI POS
0308 B1 28    LOOPY          LDA (BASL),Y    ;GET A SCREEN CHR
030A 29 7F                   AND #$7F        ;REMOVE 7TH BIT
030C 20 1F 03                JSR BYTEOUT     ;SEND TO PRINTER
030F C8                      INY
0310 C0 28                   CPY #40         ;SEND 40 CHRS
0312 D0 F4                   BNE LOOPY
0314 20 1D 03                JSR LF          ;DO A LINEFEED
0317 E8                      INX
0318 E0 18                   CPX #24         ;PRINT 24 LINES
031A D0 E6                   BNE LOOPX
031C 60                      RTS
031D          *
031D A9 0A    LF             LDA #$0A        ;LINEFEED CODE
031F 2C C1 C1 BYTEOUT        BIT $C1C1       ;TEST IF PRINTER IS BUSY
0322 30 FB                   BMI BYTEOUT     ;YES, THEN WAIT
0324 8D 90 C0                STA $C090       ;NO, THEN SEND
0327 60                      RTS
0328                         END

***** END OF ASSEMBLY
```

列表三

```
5  PRINT : INPUT "USE WHICH PRINT
     SHOP GRAPHIC ?";PS$
6  PRINT  CHR$ (4);"BLOAD ";PS$;"
     ,A$9000": FOR I = 1 TO 44: POKE
     37436 + I,0: NEXT
10  IF  PEEK (768) <  > 169 THEN
      PRINT  CHR$ (13); CHR$ (4);
      "BLOAD PSA.OBJ"
11 AD = 36864
12  GOSUB 100
15  FOR I = 1 TO 7
18  POKE 7, INT (AD / 256): POKE
      6,((AD / 256) -  INT (AD / 2
      56)) * 256
20  POKE 0,27: CALL 772
30  POKE 0,76: CALL 772
40  POKE 0,88: CALL 772
50  POKE 0,0: CALL 772
60  CALL 791
70  POKE 0,10: CALL 772
75 AD = AD + 11 * 8
80  NEXT
90  END
100  POKE 0,27: CALL 772
110  POKE 0,65: CALL 772
120  POKE 0,7: CALL 772
130  RETURN
```

列表四

```
0800           **************************
0800           *                        *
0800           *        PSA.OBJ         *
0800           *                        *
0800           *  COPYRIGHT (C) 1985    *
0800           *                        *
0800           *   BY COMPUTING AGE     *
0800           *                        *
0800           * AUTHOR:WONG WAI KONG   *
0800           *        1-8-85          *
0800           *                        *
0800           **************************
0800           *
0800           * ZERO PAGE USAGE
0800           *
0006           BUFL      EPZ $06
0007           BUFH      EPZ $07
00FF           PTN       EPZ $FF
00FD           NO        EPZ $FD
00FE           NOH       EPZ $FE
00FB           COUNT     EPZ $FB
0000           DATA      EPZ $00
0800           *
0800           * MONITOR SUBROUTINES
0800           *
0800           *
0800           * PSA.OBJ START HERE
0800           *
0300                     ORG $300
0300                     OBJ $300
0300           *
0300 A9 0A     LF        LDA #10
0302 85 00               STA DATA
0304 A5 00     BYTEOUT2  LDA DATA         ;SEND DATA TO PRINTER
0306 2C C1 C1  BYTEOUT   BIT $C1C1        ;SEE IF PRINTER IS BUSY
0309 30 FB               BMI BYTEOUT      ;YES, THEN WAIT
030B 8D 90 C0            STA $C090        ;NO, THEN SEND
030E 60                  RTS
030F           *
030F           *
030F 00        B0        HEX 00           ;BUFFER AREA
0310 00        B1        HEX 00           ;FOR INTERNAL WORKING
0311 00        B2        HEX 00
0312 00        B3        HEX 00
0313 00        B4        HEX 00
0314 00        B5        HEX 00
0315 00        B6        HEX 00
0316 00        B7        HEX 00
0317           *
0317 A9 00     APSLINE   LDA #$00         ;PRINT 1/7 PRINTSHOP GRAPHIC
0319 85 FB               STA COUNT
031B A2 00     LOOP2     LDX #$00         ;INITIALIZE FOR LOOPING
031D A0 00               LDY #$00
031F B1 06     LOOP3     LDA (BUFL),Y     ;STORE DATA TO BUFFER
0321 9D 0F 03            STA B0,X         ;BY INDIRECT ADDRESSING
0324 E8                  INX
0325 98                  TYA
```

```
0326 18                  CLC
0327 69 0B               ADC #$0B
0329 A8                  TAY
032A E0 08               CPX #$08
032C D0 F1               BNE LOOP3
032E A2 00               LDX #$00
0330 A9 00     LOOP4     LDA #$00          ;GENERATE A PRINTABLE BYTE
0332 2E 0F 03  BIT0      ROL B0
0335 90 02               BCC BIT1
0337 09 80               ORA #$80
0339 2E 10 03  BIT1      ROL B1
033C 90 02               BCC BIT2
033E 09 40               ORA #$40
0340 2E 11 03  BIT2      ROL B2
0343 90 02               BCC BIT3
0345 09 20               ORA #$20
0347 2E 12 03  BIT3      ROL B3
034A 90 02               BCC BIT4
034C 09 10               ORA #$10
034E 2E 13 03  BIT4      ROL B4
0351 90 02               BCC BIT5
0353 09 08               ORA #$08
0355 2E 14 03  BIT5      ROL B5
0358 90 02               BCC BIT6
035A 09 04               ORA #$04
035C 2E 15 03  BIT6      ROL B6
035F 90 02               BCC BIT7
0361 09 02               ORA #$02
0363 2E 16 03  BIT7      ROL B7
0366 90 02               BCC SKIP2
0368 09 01               ORA #$01
036A 20 06 03  SKIP2     JSR BYTEOUT       ;SEND THE BYTE TO PRINTER
036D E8                  INX
036E E0 08               CPX #$08
0370 D0 BE               BNE LOOP4
0372 E6 06               INC BUFL
0374 D0 02               BNE SKIP1
0376 E6 07               INC BUFH
0378 E6 FB     SKIP1     INC COUNT
037A A5 FB               LDA COUNT
037C C9 0B               CMP #$0B
037E D0 9B               BNE LOOP2         ;LOOPING UNTIL 88 BYTES SENDED
0380 60                  RTS
0381           *
0381 A0 D0     SOUND     LDY #$D0          ;GENERATE A KICK SOUND
0383 A9 08     LOOP5     LDA #$08
0385 20 A8 FC            JSR $FCA8
0388 AD 30 C0            LDA $C030
038B 88                  DEY
038C D0 F5               BNE LOOP5
038E 60                  RTS
038F           *
038F A9 58     INV       LDA #$58          ;INVERT A PRINTSHOP GRAPHIC
0391 85 07               STA BUFH          ;I.E. BLACK TO WHITE
0393 A9 00               LDA #$00          ;AND WHITE TO BLACK
0395 85 06               STA BUFL          ;THEORY : USE 255 TO SUBTRACT
ALL BYTES
0397 A8                  TAY               ;THEN ALL BITS WILL BE INVERTE
D
0398 A9 FF     LOOP1     LDA #$FF          ;E.G. 1111111-10101010 = 0101
0101
039A 38                  SEC
039B F1 06               SBC (BUFL),Y
039D 91 06               STA (BUFL),Y
039F E6 06               INC BUFL
03A1 D0 F5               BNE LOOP1
03A3 E6 07               INC BUFH
03A5 A5 07               LDA BUFH
03A7 C9 61               CMP #$61
03A9 D0 ED               BNE LOOP1
03AB 60                  RTS
03AC                     END

***** END OF ASSEMBLY
```

中文咭
應用專欄

從程式中學習編寫中文咭

程式技巧——砌唐詩遊戲

□謝秉雄□

在上幾期的電腦時代裏，我們已把中文卡的一般運作和技術資料研究過了。在今期開始我們將會以利用中文卡來編寫BASIC 的程式爲我們研究對像。

在開始時，先讓我們看一看電腦的基本結構：基本上來說，電腦是一部機器，它會依着人的指示去做事，電腦沒有耳朵，所以你不能對著部電腦大聲叫：〔一加一是多少？〕令電腦明白你想叫它做什麽而把答案告訴你。要叫電腦明白你的指示便要把資料從鍵盤中輸入至電腦，而電腦才可以根據你的指示作運算，然後把結果輸出至螢光幕上去。由此可見，輸出與輸入，在電腦的操作中是極其重要。要寫一篇好的BASIC 程式一定要對輸出、入有較全面的認識。

圖(1)列出的程式是一個砌唐詩的遊戲，遊戲開始時會印出一首唐詩，然後問你難度是多少，這時你可以隨意鍵入一個數目字，電腦便會跟據數字的次序把詩句裏的字弄亂，之後你便要用I、J、K或M鍵把反白的一字在詩句中向上、下、右和左移動，電腦會把字句互調，直到你把詩調回原來的樣子，便完成了這遊戲。

在這程式裏我用上了一些特別的方法處理畫面的輸出，使程式加添色彩。

中文卡是利用HGR或HGR2 來做畫面的，HPLOT，HPLOT TO 等高解像用的命令在中文卡的畫面中是可以正常操作，在程式的100至109行便是利用了這些指令來把畫面畫出。

在中文卡的BASIC 之中是沒有FLASH (閃動)的，但利用一些技巧亦可以做到，辦法是不斷的利用一個黑底白字的句子和一句白底黑字的句子，在同一位置不停的互相替換便可，在程式中的第112行便是利用這種技術把名字在畫面中閃動。

在程式中把兩個字互調的方法是利用VTAB和HTAB而達成。

電腦的輸出除了畫面外還有喇叭，要電腦發出"嗶"聲，可以利用PRINT CHR $(7)。在1510行我先把CHR $(7)放入在G $內，然後輸出一個G $，電腦便會發出一聲"嗶"，就像開機時一樣。

程式中已有REM解譯程式的運作，讀者如果能完全明白當然最好，但最重要的是要知道電腦如何輸出畫面，把握到這些技術後便可以在自己的程式中加以運用編寫出一篇出色的中文程式。

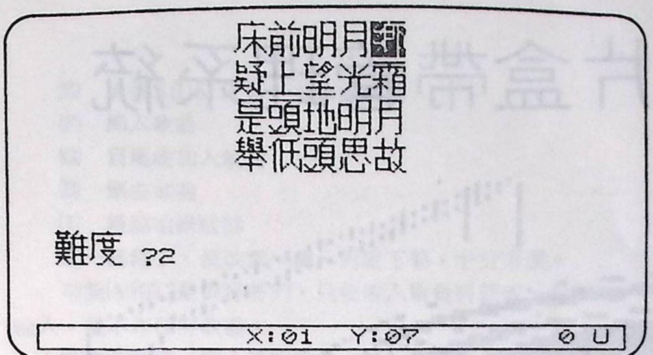
各位讀者如果有一些爲中文卡而編寫的程式，可以向電腦時代投稿，一同分享。同時亦歡迎各讀者來信一同研究一些在中文卡上的問題，反映你們想要知道的題目。■

```
1   HIMEM:  16384
10  HOME
20 G$ =  CHR$ (7): REM  嗶聲
30 T$ = "作者：謝秉雄"
100  HTAB (12): VTAB (10): P
RINT T$
101  REM  劃出高解像圖案
102  FOR F = 22 TO 150 STEP
4
104  HPLOT 170,1 TO 1,F
105  HPLOT 170,180 TO 1,F
106  HPLOT 87,1 TO 255,F
107  HPLOT 87,180 TO 255,F
109  NEXT
110  REM  將作者名閃動
112  FOR F = 1 TO 10: INVERS
E : HTAB (12): VTAB (10): PR
INT T$;: NORMAL : HTAB (12):
 VTAB (10): PRINT T$;: NEXT
130  REM  開始遊戲
137  HOME
140 R = 1
145  REM  把詩句從 DATA 中讀
入
150  READ C$(R)
155  IF C$(R) = "*" THEN 300
156  REM  把詩句中的字分別放
入 A$(R,F) 中( R 是行數 F 是
字數 )
157 C$(R) = "金" + C$(R)
159 C =  LEN (C$(R)) / 5
160  FOR F = 2 TO C
170 A$(R,F - 1) =  MID$ (C$(
R),(F - 1) * 5) + 1,5)
177  NEXT
180 R = R + 1
190  GOTO 150
200  REM  詩的 DATA
```

從程式中學習編寫中文咭程式技巧── 砌唐詩遊戲

編者註：我們懇切徵求各類專爲利用中文咭而編寫的文章及程式，或者閣下如有興趣跟我們一齊研究編寫中文軟件或有興趣嘗試編寫中文程式，請立即利用熱線電話3-312007跟我們聯絡，安排約見。

要輸入本程式，大家必須擁有一張可執行「超級中文DOS」（即CCDOS）的中文咭，與及在CCDOS下鍵入。輸入的方法一如普通APPLESOFT 程式，只是中文字部份大家必須要利用倉頡輸入法中文字。完成鍵入後可用SAVE MOON 指令存入磁碟，或者各位可在即期出版的程式磁碟中找到相同名字的檔案。大家RUN 此程式時必須要在CCD－OS 記憶系統中。

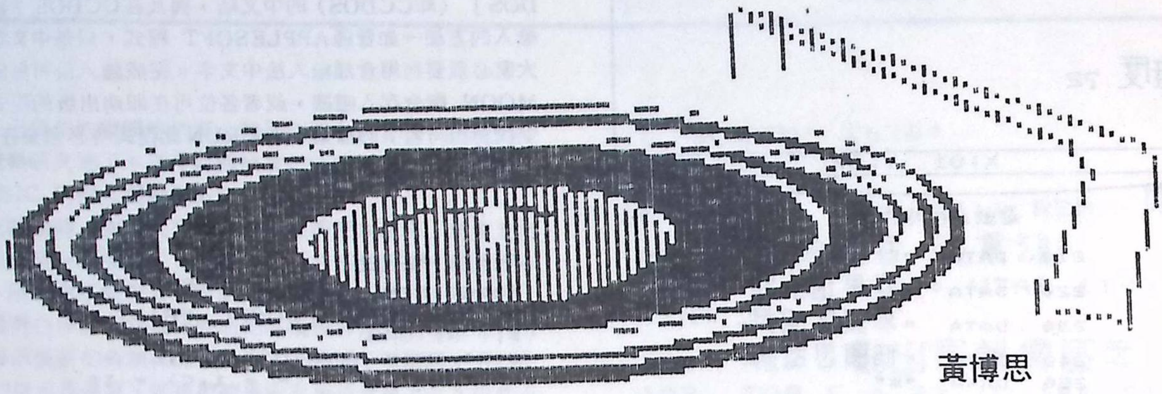


遊戲進行時的畫面

```
210  DATA "床前明月光"
220  DATA "疑是地上霜"
230  DATA "舉頭望明月"
240  DATA "低頭思故鄉"
250  DATA "*"
300 C = C - 1:R = R - 1
305 X = C:Y = R
310  GOSUB 1000
312  GOSUB 6000
314 FL = 1: REM  FL=0 時代表
完成
350  REM 印出反白小塊
355  INVERSE
360  VTAB (R * 2): HTAB (C *
 2 + 10): PRINT A$(R,C);: NO
RMAL
365  IF FL = 0 THEN  GOTO 50
0
370  PRINT : PRINT : GET Y$
380  IF Y$ = "I" THEN  GOSUB
 2000
390  IF Y$ = "M" THEN  GOSUB
 3000
400  IF Y$ = "J" THEN  GOSUB
 4000
410  IF Y$ = "K" THEN  GOSUB
 5000
420  GOSUB 7000
430  GOTO 355
500  VTAB (16): HTAB (1): PR
INT " 很好!! 再來一次嗎? ";:
 GET T$
510  IF T$ = "Y" THEN  RUN
520  PRINT "再見 !!": GET T$
: RUN
1000  REM  把詩印出
1010  FOR F = 1 TO R
1020  FOR G = 1 TO C
1030  VTAB (F * 2): HTAB (G
* 2 + 10): PRINT A$(F,G)
1040  NEXT
1050  NEXT
1060  RETURN
2000  REM  向上移一步
2010  IF R = 1 THEN  PRINT G
$;: RETURN
2020 T$ = A$(R,C):A$(R,C) =
A$(R - 1,C):A$(R - 1,C) = T$
2030  VTAB (R * 2): HTAB (C
* 2 + 10): PRINT A$(R,C)
2040 R = R - 1
2050  RETURN
3000  REM  向下移一步
3010  IF R = Y THEN  PRINT G
$;: RETURN
3020 T$ = A$(R,C):A$(R,C) =
A$(R + 1,C):A$(R + 1,C) = T$
3030  VTAB (R * 2): HTAB (C
* 2 + 10): PRINT A$(R,C)
3040 R = R + 1
3050  RETURN
4000  REM  向左移一步
4010  IF C = 1 THEN  PRINT G
$;: RETURN
4020 T$ = A$(R,C):A$(R,C) =
A$(R,C - 1):A$(R,C - 1) = T$
4030  VTAB (R * 2): HTAB (C
* 2 + 10): PRINT A$(R,C)
4040 C = C - 1
4050  RETURN
5000  REM  向右移一步
5010  IF C = X THEN  PRINT G
$;: RETURN
5020 T$ = A$(R,C):A$(R,C) =
A$(R,C + 1):A$(R,C + 1) = T$
5030  VTAB (R * 2): HTAB (C
* 2 + 10): PRINT A$(R,C)
5040 C = C + 1
5050  RETURN
6000  REM  根擄難度把詩句弄
亂
6005  VTAB (16): HTAB (2): I
NPUT "難度 ?";N
6010  FOR F = 1 TO N
6020 V =  INT ( RND (1) * 10
)
6030  IF V > 3 THEN 6020
6040  ON V + 1 GOSUB 2000,30
00,4000,5000
6045  NEXT
6050  RETURN
7000  REM  測試是否完成
7002 FL = 0
7005  FOR F = 1 TO R
7007 T$ = "金"
7010  FOR G = 1 TO C
7020 T$ = T$ + A$(F,G)
7030  NEXT
7040  IF T$ <  > C$(F) THEN
FL = 1
7050  NEXT
7060  RETURN
```

一個專為唱片盒帶擁有者度身訂造的

dBASE 程式——唱片盒帶管理系統



黃博思

相信閣下家中也珍藏了不少唱片和盒帶，有沒有試過想聽一首歌時，却記不起它放在那張碟或帶內的掃興事呢？又想不想在極短的時間內知道自己收藏了一位歌手的作品有多少？又或者數數自己擁有幾多隻十二吋大碟？

以上種種，今期的唱片管理系統全部可以爲你辦到。市面上的資料管理系統程式雖然衆多，但畢竟不是「度身訂造」的，強大的先要建立一個資料庫，命令檔等一大串，加上唱片的紀錄欄包括唱碟名稱，樂手名稱，唱碟類型，樂曲名稱，有幾多首歌，那兒是SIDE B等，一欄和另一欄各有不同關係，有別於一般的電話人事紀錄，或者郵遞記錄等。「唱片管理系統」就是一個針對以上資料欄而寫的友善易用專業程式。在程式執行過程中，有足夠的螢幕程式操作指示，閣下即使對電腦程序毫無認識，也能完全掌握這程序的使用。

程式須在最少一台碟機的APPLE 上運行，亦可匹配印字機和第二台碟機，方便有多個唱片記錄的用者。記憶的容量當然是越大越好了。

程序功能選擇

程式的FILENAME是RECORDING DBASE，一開始你便會看到一個程序功能選表，選表上有七項主要程序功能，其分別順序如下：

(1) ENTER NEW RECORD (輸入新記錄)
(2) EDIT RECORD CONTENT (編輯記錄內容)
(3) DELETE A RECORD (刪去一項記錄)
(4) DUMP RECORD CONTENT (傾印紀錄內容)
(5) SEARCH A RECORD (找尋一項記錄)
(6) DISK VTILITIES (使用磁碟)
(7) QUIT PROGRAM (離開程式)

需要運用的鍵只有四個，用左右箭咀能把反白光條移至選擇項目上，再按RETURN 便可進入該項功能。在選表中按ＥＳＣ便會令反白光條跳到「離開」一項上面，在其他功能內，ＥＳＣ鍵也有「離開」的功能。在操作時按CTRL－C便可返回此程序選表。以下將功能１－６順序逐項介紹。

輸入新記錄

電腦會詢問唱片盒帶的名稱，若你不作任何輸入，便代表放棄此項功能，程序便跳回選表去。否則便須繼續順序輸入唱碟的類別（大碟，十二吋混音，七吋細碟），樂手姓名，有幾多首歌，和逐一順序輸入歌曲名稱。唱碟類型爲了方便用者，可用左右箭咀和RETURN 選擇答案，程序提供的唱片類型只有四個，大碟，十二吋混音，七吋細碟，和圖案碟，若要改變或增加唱碟類型，可參閱程序改動部份。另外，爲了操作方便，唱碟和歌曲都被冠上一個編號，最後你需要輸入SIDE B 第一隻歌的編號了。

若以上的輸入無誤，可按RETURN 返回選表，否則可按ＥＳＣ進入編輯系統。

編輯系統

這項功能可以在「程序功能表」中選取，亦可以在執行其他功能如上面的「輸入新記錄」和「傾印記錄內容」時，以某些功能鍵直接跳到這裏，這樣便讓你一發現錯誤，便可儘快修改，更爲直接方便。

若此項功能是從其他功能表中進入的，便讓你編輯原先在處理中的記錄。若是從選表中進入的，電腦便詢問你唱片的名稱，或唱片的編號，可隨意輸入兩者之一便可，非常易用。

螢幕隨即印出一多項功能選擇表讓你揀選欲修改的資料欄，選擇鍵是左右箭咀，RETURN和ESC 。

選表順序分別如下：

(A) 編輯唱片名稱
(B) 編輯唱片類型
(C) 編輯樂手名稱
(D) 編輯歌曲名稱

唱片盒帶管理系統

(E) 編輯SIDE B 第一首歌曲編號
(F) 插入歌曲
(G) 自尾端加入歌曲
(H) 刪去歌曲
(I) 離開編輯狀態

那一欄錯了，便改那一欄，對症下藥，十分方便。

功能(A)和(C)是很容易的，只要輸入新資料便成，若不作輸入，便不作任何改動。

功能(B)和在「輸入新記錄」一樣，用左右箭咀選了類型，再按RETURN 便可。

功能(G)只要鍵入新加入的歌曲名稱便可。

功能(D)，(E)，(F)，(H)在執行時先在螢幕印出唱片盒帶內的所有歌曲，再讓你選擇欲處理的歌曲，選擇方法和功能選表一樣。選擇了歌曲，若是功能(F)或(D)，電腦便會詢問你歌曲的名稱。

刪去一項記錄

這項功能讓你刪去記錄，電腦會要求你按Y證實。若這項功能是從選表中進入，你便須要鍵入記錄的名稱或編號，否則若是從「傾印記錄內容」中進入的，便刪去當時正在螢幕顯示的唱片記錄。

傾印記錄內容

此項功能可說是整個程式的心臟，它把幾個功能和傾印功能連起來，可以讓你翻查審閱資料，螢幕上亦會出現另一個多項功能選擇表。其內容順序分別如下：

(A) 印出所有唱片盒帶內容
(B) 印出某一樂手的作品
(C) 印出某一類型的唱片盒帶
(D) 改變輸出媒介
(E) 離開印出記錄功能

印出所有記錄，若以視訊媒介作輸出的話，在螢幕下端便會有下列的功能鍵顯示。記錄資料會以逐頁形式印出，一頁一隻碟。以下的功能應用非常方便，只要鍵入括號中的鍵，便立即執行。

〔←〕左箭頭，倒捲資料頁顯示。
〔→〕右箭頭，快速前捲，可飛至某一資料頁。
〔E〕編輯記錄，可直接跳到編輯系統，編輯目前的資料頁。
〔D〕刪去目前顯示的記錄，即跳去刪除記錄功能去。
〔ESC〕 跳回編印功能選表。
〔P〕用印字機印出目前在螢幕顯示的記錄。

執行印出某一樂手的作品時，電腦會詢問樂手的名字，然後順序印出他的作品。

執行印出某類型的唱片也是差不多，用左右箭咀揀選了類型，按RETURN 鍵，電腦便印出該類型的唱片了。

輸出媒介（TERMINAL） 是指螢光幕或印字機兩個選擇，要改變現在使用的輸出媒介，只要移到項目上按RETURN，你會看到螢光幕頂部的「目前使用的媒介」，由視訊變為印字機，或由印字機變為視訊，再在項目上按RETURN，又可變回原先的媒介。

從記錄中找出歌曲

找尋這項功能是在一切資料管理程序中都不會缺少的，本程式的找尋功能是分開幾個程度，除最基本的資料——要找的歌曲外，電腦會問你能否提供輔助資料的，問題要求的是一個「正負」的答案（Y／N），而DEFAULT 的答案是否定的，你可以用RETURN或SPCALE略去答覆。若你能提供輔助資料，便需作有關輸入，輔助資料是樂手名稱和歌曲版本，它們可以節省比較資料的次數，當然單靠歌名亦可知道它在那一張碟，由何樂手負責，是何版本，在那一面的那一首歌，但找尋就比較費時了。無論找到歌曲與否，都會讓你知道找尋結果的。

使用磁碟

這項功能亦提供以下的功能選表，其順序如下：

(A) 改變磁碟機號
(B) 讀取唱片記錄
(C) 儲存唱片記錄
(D) 刪除唱片記錄
(E) 在印字機上印出磁碟目錄
(F) 離開磁碟使用功能

功能(A)相信你在TAKE 1和PIXIT 都見過，功能(B)到(D)是需要輸入FILENAME 的，若想放棄執行的話，可以不作任何的輸入，便會返回功能選表的了。

由於這程式是以用家友善為原則的，當中有多處是可以讓你裁剪修改到貼身訂造的，其順行號次序如下；另外，在程式列表內有REM作簡單標示的。

程序改動

(1)行110 的PS＝1是印字機的SLOT NO. ，用者的PRINTER 若在其他SLOT，便須改PS 值。

(2)行120 的D＝1是DEFAULT 磁碟機號。若你習慣用DRIVE 2 做資料存取的話，改了D值會更方便。

(3)行130 的VN＝ 20是記錄的容量，即可容納多少唱碟，在記憶許可的情況下，此值應越大越好。

(4)行140 的SN＝20 是每隻碟或盒帶可容納最多的歌曲。

(5)行200 的RN＝4 和跟着的STRING 是有關唱片類型的，RN 是有幾多類型，STRING 是類型名稱。

(6)行2190的1F ER＝255 THEN 330 把CTRL－C 鍵的中止功能鎖住，讓你可以隨時返回選表，只要刪去此行，便可以隨時中止程式的操作了。

程式是絕對「度身定做」的，由今天起，就把唱片記錄電腦化吧！ ■

```
10  REM ************************
20  REM * RECORD MASTER DBASE. *
30  REM * FOR ALBUM,12" SINGLE *
40  REM * 7" SINGLES...ETC.    *
50  REM * AND EVEN TAPES...    *
60  REM * AUTHOR: WONG POK SZE *
70  REM *    COMPUTING AGE     *
80  REM ************************
90  REM
100 CLEAR : TEXT : NORMAL : SPEED=
    255
110 PS = 1: REM  PRINTER SLOT ***
120 D = 1: REM  DEFAULT DRIVE ***
130 VN = 20: REM  RECORD VOLUME *
    **
140 SN = 20: REM  SONG NO. PER RE
    CORD. ***
150 REM  DEF. VARIABLES ***
160 D$ =  CHR$ (13) +  CHR$ (4): REM
     CTRL-D ***
170 FOR I = 1 TO 40:LN$ = LN$ +
    "-":L$ = L$ +  CHR$ (95): NEXT
    :LL$ =  LEFT$ (L$,30):B$ =  CHR$
    (7) +  CHR$ (7) +  CHR$ (7):
    P$(PS) = "PRINTER":P$(0) = "
    VISUAL "
180 DIM S$(VN,SN),A$(VN),AR$(VN)
    ,T(VN),C(VN),F(VN)
190 REM  TYPES OF RECORD ***
200 RN = 4:C$ =  CHR$ (34):R$(1) =
    "12" + C$ + " SINGLE":R$(2) =
    "7" + C$ + " SINGLE":R$(3) =
    "ALBUM":R$(4) = "PICTURE DIS
    K"
210 I(0) = 7: FOR I = 1 TO I(0): READ
    I$(0,I),I(I): NEXT : DATA "E
    NTER NEW RECORD",1,"EDIT REC
    ORD CONTENT",9,"DELETE A REC
    ORD",1,"DUMP RECORD CONTENT"
    ,5,"SEARCH A SONG FROM RECOR
    D",1,"DISK UTILITIES",6,"QUI
    T PROGRAM",1
220 REM  READ OPTION DATA ***
230 FOR I = 1 TO I(0): FOR J = 1
     TO I(I): READ I$(I,J): NEXT
    J,I
240 DATA ""
250 DATA "EDIT RECORD NAME","EDI
    T RECORD TYPE","EDIT ARTIST
    NAME","EDIT SONG NAME","EDIT
     FLIP SIDE NO.","INSERT SONG
     AT LIST","ADD SONG AT LIST"
    ,"DELETE SONG AT LIST","QUIT
     EDITOR"
260 DATA ""
270 DATA "PRINT ALL RECORD","PRI
    NT RECORD OF AN ARTIST","PRI
    NT RECORD OF ONE TYPE","CHAN
    GE PRINT TERMINAL","QUIT PRI
    NT SUBSYSTEM"
280 DATA ""
290 DATA "CHANGE DRIVE","LOAD RE
    CORD FILE","SAVE RECORD FILE
    ","DELETE RECORD FILE","DUMP
     CATALOG TO PRINTER","QUIT D
    ISK UTILITIES"
300 DATA  ""
310 REM  ***************
320 REM  ** MAIN MENU **
330 REM  ***************
340 TEXT : HOME : PRINT "RECORD
    FILE MASTER        BY WONG PO
    K SZE": VTAB 3: HTAB 1: PRINT
    "MAIN MENU"
350 T = 0: GOSUB 360: IF V > 1 AND
    V < 6 AND N < 1 THEN  HTAB 1
    : VTAB 22: PRINT B$" NO RECO
    RDING. DATA IN MEMORY"; TAB(
    39): GOTO 2310
355 ON V GOTO 470,630,1110,1240,
    1720,1960,2180
360 VO = 4:UB = I(T): POKE 34,4: POKE
    35,24: HOME : HTAB 1: VTAB 5
    : FOR I = 1 TO UB: PRINT I$(
    T,I): NEXT : PRINT : PRINT :
     PRINT "[<-].[->] SELECT..[
    RETURN] EXECUTE":V = 1
370 VTAB VO + V: HTAB 1: INVERSE
    : PRINT I$(T,V): NORMAL
380 K =  PEEK ( - 16384): IF K <
    128 THEN 380
390 POKE  - 16368,0: HTAB 1: VTAB
    VO + V: PRINT I$(T,V): IF K =
    136 THEN V = V - 1: IF V < 1
    THEN V = UB
400 IF K = 141 THEN  RETURN
410 IF K = 149 THEN V = V + 1: IF
    V > UB THEN V = 1
420 IF K = 155 THEN V = UB
430 GOTO 370
440 REM  ******************
450 REM  ** INIT A RECORD **
460 REM  ******************
470 POKE 34,0: HOME : PRINT "INI
    T A RECORD..."
480 HTAB 1: VTAB 3: INPUT "NAME
    OF RECORD :";A$: IF A$ = "" THEN
    340: REM  GOTO MENU
490 N = N + 1:A$(N) = A$
500 V = 5: GOSUB 810:T(N) = I
510 HTAB 1: VTAB 7: PRINT  TAB(
    39): HTAB 1: VTAB 7: INPUT "
    NAME OF ARTIST (IF ANY) :";A
    R$(N)
520 HTAB 1: VTAB 9: INPUT "NO. O
    F CONTENT :";C(N)
530 VTAB 11: PRINT "ENTER SONG N
    AME...": POKE 34,12: VTAB 13
    : FOR I = 1 TO C(N): HTAB 2:
     PRINT I; TAB( 4);".....";: INPUT
    " ";S$(N,I): NEXT
540 HTAB 1: VTAB 22: INPUT "FLIP
     SIDE :A->B AT NO :";A$: IF
    A$ = "" THEN 540
550 F(N) =  VAL (A$)
560 VTAB 19: HTAB 24: PRINT  TAB(
    39);: VTAB 20: HTAB 24: PRINT
    " [RETURN] IF DATA": VTAB 21
    : HTAB 24: PRINT " READY....
    ": VTAB 22: HTAB 24: PRINT "
    [ESC] FOR EDITOR"
570 K =  PEEK ( - 16384): IF K <
     > 141 AND K <  > 155 THEN 5
    70
580 POKE  - 16368,0: IF K = 141 THEN
    340: REM  MENU ***
590 E = N: GOTO 640
```

```
600  REM  *****************
610  REM  ** THE EDITOR **
620  REM  *****************
630  POKE 34,0: HOME : PRINT "THE
      EDITOR...": GOSUB 1150
640  VTAB 3: HTAB 1: PRINT "RECOR
     D TO BE EDIT :";A$(E);"  (";
     E;")"; TAB( 39)
650 T = 2: GOSUB 360: POKE 34,4: HOME
     : ON V GOSUB 770,800,890,930
     ,980,1010,1050,1080,340: GOTO
     650
660  REM   SONG LISTING ***
670  VTAB 5: PRINT "A: ";: FOR I =
     1 TO F(E): HTAB 4: PRINT S$(
     E,I): NEXT
680 V =  PEEK (37): VTAB V: HTAB
     1: PRINT "B: ";: FOR I = F(E
     ) TO C(E): HTAB 4: PRINT S$(
     E,I): NEXT :VO = 4:V = 1
690  HTAB 4: VTAB VO + V: INVERSE
     : PRINT S$(E,V): NORMAL
700 K =  PEEK ( - 16384): IF K <
     128 THEN 700
710  POKE  - 16368,0: VTAB VO + V
     : HTAB 4: PRINT S$(E,V): IF
     K = 136 THEN V = V - 1: IF V
      < 1 THEN V = C(E)
720  IF K = 141 THEN  INVERSE : HTAB
     4: VTAB VO + V: PRINT S$(E,V
     ): NORMAL : RETURN
730  IF K = 155 THEN 650
740  IF K = 149 THEN V = V + 1: IF
     V > C(E) THEN V = 1
750  GOTO 690
760  REM   EDIT RECORD NAME ***
770  INPUT "ENTER NEW RECORD NAME
     :";A$: IF A$ = "" THEN 640
780 A$(E) = A$: GOTO 640
790  REM   EDIT RECORD TYPE ***
800 V = 5: GOSUB 810:T(E) = I: RETURN

810 I = 1: VTAB V: PRINT "TYPES O
     F RECORD :": HTAB 1: VTAB V +
     2: PRINT "[<-].[->] SELECT..
     .[RETURN] CONFIRM"
820  HTAB 20: VTAB V: PRINT R$(I)
     ; TAB( 39)
830 K =  PEEK ( - 16384): IF K <
     128 THEN 830
840  POKE  - 16368,0: IF K = 136 THEN
     I = I - 1: IF I < 1 THEN I =
     RN: GOTO 820
850  IF K = 149 THEN I = I + 1: IF
     I > RN THEN I = 1: GOTO 820
860  IF K = 141 THEN  VTAB V + 3:
      RETURN
870  GOTO 820
880  REM   EDIT ARTIST NAME ***
890  PRINT "CURRENT ARTIST VALUE
     :";AR$(E): PRINT
900  INPUT "NEW VALUE :";A$: IF A
     $ = "" THEN  RETURN
910 AR$(E) = A$: RETURN
920  REM   EDIT SONG NAME ***
930  HTAB 1: VTAB 20: PRINT "SELE
     CT ITEM TO BE EDIT": GOSUB 6
     70
940  HTAB 1: VTAB 20: PRINT  TAB(
     39): VTAB 20: HTAB 1: INPUT
     "ENTER NEW VALUE :";A$
950  IF A$ = "" THEN 930
960 S$(E,V) = A$: GOTO 930
970  REM   EDIT FLIP NO. ***
980  POKE 34,4: HOME : HTAB 1: VTAB
     20: PRINT "POSITION CUSOR AT
      SIDE B,1st SONG": GOSUB 670
990 F(E)= V: RETURN
1000  REM   INSERT ***
1010  POKE 34,4: HOME : HTAB 1: VTAB
     20: PRINT "POSITION CUSOR AT
      INSERT": GOSUB 670
1020  VTAB 22: PRINT V: FOR J = C
     (E) TO V STEP  - 1:S$(E,J +
     1) = S$(E,J): NEXT :C(E) = C
     (E) + 1
1030  HTAB 1: VTAB 20: PRINT  TAB(
     39): VTAB 20: HTAB 1: INPUT
     "ENTER VALUE :";S$(E,V): GOTO
     1010
1040  REM   ADD AT END ***
1050  INPUT "ADD AT END OF RECORD
     :";A$: IF A$ = "" THEN  RETURN

1060 C(E) = C(E) + 1:S$(E,C(E)) =
     A$: RETURN
1070  REM   DELETE ***
1080  POKE 34,4: HOME : HTAB 1: VTAB
     20: PRINT "SELECT ITEM TO BE
      ";: INVERSE : PRINT "DELETE
     ": NORMAL : GOSUB 670
1090  FOR I = V TO C(E):S$(E,I) =
     S$(E,I + 1): NEXT :C(E) = C(
     E) - 1: GOTO 1080
1100  REM   *********************
1110  REM  ** DELETE A RECORD **
1120  REM  *********************
1130  HOME : GOSUB 1150
1140  VTAB 3: HTAB 1: PRINT "RECO
     RD TO BE ";: INVERSE : PRINT
     "DELETE";: NORMAL : PRINT "
     :";A$(E);"  (";E;")"; TAB( 3
     9):I = E: GOSUB 1500: GOTO 3
     40
1150  HTAB 1: VTAB 3: INPUT "NAME
      OR CODE OF RECORD :";A$: IF
     A$ = "" THEN 340: REM  MENU
     ***
1160 E =  VAL (A$): IF E < > 0 AND
     E = < N THEN  RETURN
1170  FOR I = 1 TO N: IF A$(I) =
     A$ THEN E = I
1180  NEXT
1190  IF E = 0 THEN  HTAB 1: VTAB
     5: PRINT "NO SUCH RECORD EXI
     STS.";: GOSUB 2200: HTAB 1: PRINT
      TAB( 39): GOTO 1150
1200  RETURN
1210  REM  *********************
1220  REM  ** PRINTER SYSTEM **
1230  REM  *********************
1240  POKE 34,0: HOME : PRINT "RE
     CORD DUMP SUBSYSTEM...": VTAB
     3: HTAB 1: PRINT "CURRENT TE
     RMINAL :";P$(PL):T = 4: GOSUB
     360: ON V GOSUB 1260,1530,16
     00,1670,340: GOTO 1240
1250  REM  DUMP ALL RECORD ***
1260  IF PL = PS THEN  PRINT D$;"
     PR#";PS: PRINT : PRINT : PRINT
     "PRINT ALL RECORD FILE...": PRINT
     : PRINT : FOR I = 1 TO N: GOSUB
     1300: PRINT : PRINT : NEXT :
     PRINT D$;"PR#0": GOTO 1240
1270 I = 1: POKE 34,0: POKE 35,23
     : HOME : PRINT "DUMP ALL REC
     ORDS": PRINT LN$
```

```
1280  VTAB 20: HTAB 2: PRINT "[<-
      ] F.REW    [E]=EDIT    [P]=
      PRINTER": HTAB 2: PRINT "[->
      ] FFWD     [D]=DELETE   [ESC
      ]=MENU": POKE 34,2: POKE 35,
      19: HOME : GOSUB 1300: GOTO
      1340
1290  GOSUB 1300: GOTO 1330
1300  HOME : PRINT "NAME OF RECOR
      D :";A$(I): PRINT : PRINT "T
      YPE OF RECORDING :";R$(T(I))
      : PRINT : IF AR$(I) <  > "" THEN
      PRINT "NAME OF ARTIST :";AR
      $(I): PRINT
1310  PRINT "A: ";: FOR J = 1 TO
      F(I): HTAB 4: PRINT S$(I,J):
      NEXT
1320  V =  PEEK (37): VTAB V: HTAB
      1: PRINT "B: ";: FOR J = F(I
      ) TO C(I): HTAB 4: PRINT S$(
      I,J): NEXT : RETURN
1330  K =  PEEK ( - 16384): IF K <
      128 THEN P = P + 1: GOTO 142
      0
1340  POKE  - 16368,0:P = 0: HTAB
      2: VTAB 22: PRINT  TAB( 40)
1350  REM  FUNCTIONS ***
1360  IF K = 136 THEN 1480: REM
      [<-] ***
1370  IF K = 155 THEN 1240: REM
      [ESC] ***
1380  IF K = 208 THEN  PRINT D$;"
      PR#";PS: GOSUB 1300: PRINT D
      $;"PR#0": GOTO 1330: REM  [P
      ] ***
1390  IF K = 197 THEN E = I: GOTO
      640: REM  [E] ***
1400  IF K = 196 THEN  GOSUB 1500
      : GOTO 1290: REM  [D] ***
1410  IF K = 149 THEN 1450: REM
      [->] ***
1420  IF P > 100 THEN P = 0: GOTO
      1450: REM  DELAY ENDS. ***
1430  GOTO 1330
1440  REM  FFWD ***
1450  I = I + 1: IF I =  > N THEN
      I = N: VTAB 22: HTAB 1: PRINT
      TAB( 27)"<END OF FILE>"
1460  GOTO 1290
1470  REM  F.REW ***
1480  I = I - 1: IF I < 1 THEN I =
      1: VTAB 22: HTAB 2: PRINT "<
      START OF LISTING>"; TAB( 39)
1490  GOTO 1290
1500  HOME : PRINT "TYPE [Y] TO C
      ONFIRM ";: INVERSE : PRINT "
      DELETE";: NORMAL : PRINT "
      ";: GET A$: IF A$ <  > "Y" THEN
      RETURN
1510  FOR J = I TO N:A$(J) = A$(J
      + 1):AR$(J) = AR$(J + 1):T(
      J) = T(J + 1):F(J) = F(J + 1
      ):C(J) = C(J + 1): FOR K = 1
      TO C(J):S$(J,K) = S$(J + 1,
      K): NEXT K,J:N = N - 1: RETURN
1520  REM  RECORD OF AN ARTIST **
      *
1530  HOME : HTAB 1: VTAB 5: INPUT
      "NAME OF ARTIST :";A$: IF A$
      = "" THEN 1240
1540  PRINT D$;"PR#";PL
1550 E = 0: FOR I = 1 TO N: IF A$
      = AR$(I) THEN E = 1: GOSUB
      1300
1560  GOSUB 2200: PRINT : PRINT :
      NEXT : PRINT D$;"PR#0"
1570  IF E = 0 THEN  PRINT : PRINT
      "NO SUCH ARTIST EXISTS.": GOSUB
      2200: GOTO 1240
1580  VTAB 22: HTAB 27: PRINT "<E
      ND OF FILE>": GOSUB 2200: RETURN
1590  REM  RECORD OF A TYPE ***
1600  HOME :V = 5: GOSUB 810
1610  PRINT D$;"PR#";PL
1620  HOME :N1 = I
1630 E = 0: FOR I = 1 TO N: IF T(
      I) = N1 THEN E = 1: GOSUB 13
      00
1640  GOSUB 2200: PRINT : PRINT :
      NEXT : PRINT D$;"PR#0": IF
      E = 0 THEN  PRINT "NO SUCH T
      YPE OF RECORD EXISTS.": GOSUB
      2200: GOTO 1240
1650  VTAB 22: HTAB 27: PRINT "<E
      ND OF FILE>": GOSUB 2200: RETURN
1660  REM  CHANGE TERMINAL ***
1670  IF PL = PS THEN PL = 0: GOTO
      1240
1680 PL = PS: GOTO 1240
1690  REM  ******************
1700  REM  ** SEARCH A SONG **
1710  REM  ******************
1720  POKE 34,0: HOME : PRINT "SE
      ARCH FOR A SONG..."
1730  HTAB 1: VTAB 3: INPUT "NAME
      OF SONG ? ";S$: IF S$ = "" THEN
      340: REM  MENU ***
1740  HTAB 1: VTAB 3: PRINT "SEAR
      CH SONG NAME :";S$: POKE 34,
      4
1750 S1 = 0: HTAB 1: VTAB 5: PRINT
      "DO YOU KNOW THE ARTIST NAME
      ? <N>";: HTAB 32: GET A$: PRINT
      : IF A$ = "Y" THEN  HTAB 32:
      VTAB 5: PRINT A$: HTAB 1: VTAB
      7: INPUT "NAME OF ARTIST ?";
      AR$:S1 = 1: IF AR$ = "" THEN
      1750
1760 V1 =  PEEK (37) + 2:S2 = 0: HTAB
      1: VTAB V1: PRINT "DO YOU KN
      OW THE RECORD TYPE ? <N>";: HTAB
      32: GET A$: IF A$ = "Y" THEN
      VTAB V1: HTAB 32: PRINT A$:
      V = V1 + 2: HTAB 1: GOSUB 81
      0:T = I:S2 = 1
1770  IF S1 = 1 AND S2 = 1 THEN 1
      830
1780  IF S1 = 1 THEN 1850
1790  IF S2 = 1 THEN 1810
1800  FOR I = 1 TO N: GOSUB 1880:
      NEXT I: GOTO 1870
1810  FOR I = 1 TO N: IF T = T(I)
      THEN  GOSUB 1880
1820  NEXT : GOTO 1870
1830  FOR I = 1 TO N: IF AR$(I) =
      AR$ AND T(I) = T THEN  GOSUB
      1880
1840  NEXT : GOTO 1880
1850  FOR I = 1 TO N: IF AR$(I) =
      AR$ THEN  GOSUB 1880
1860  NEXT
1870  PRINT : PRINT "NO SUCH SONG
      EXISTS.": GOTO 1910
```

```
1880  FOR J = 1 TO C(I): IF S$(I,
    J) <  > S$ THEN 1920
1890  HOME : INVERSE : PRINT B$;"
    SONG IS FOUND !!! ": NORMAL
    : VTAB 7: PRINT "RECORDING N
    AME": PRINT LN$: VTAB 9: PRINT
    A$(I);" (";I;")": PRINT : PRINT
    "BY :";AR$(I): PRINT : PRINT
    "TYPE :";R$(T(I)): PRINT : IF
    J >  = F(I) THEN  PRINT "SID
    E B..SONG :";J - F(I) + 1: GOTO
    1910
1900  PRINT "SIDE A..SONG :";J
1910  PRINT : PRINT "HIT ANY KEY
    TO CONT :";: GET A$: GOTO 34
    0
1920  NEXT J: RETURN
1930  REM  *****************
1940  REM  ** DISK ACCESS **
1950  REM  *****************
1960  POKE 34,0: HOME : PRINT "DI
    SK UTILITIES...": VTAB 3: HTAB
    1: PRINT "CURRENT DRIVE NO.
    :";D
1970  T = 6: GOSUB 360: ON V GOSUB
    1990,2010,2050,2100,2140,340
    : GOTO 1960
1980  REM  CHANGE DISK ***
1990  IF D = 1 THEN D = 2: GOTO 1
    960
2000 D = 1: GOTO 1960
2010  REM  LOAD ***
2020  POKE 34,4: HOME : VTAB 19: PRINT
    LN$: POKE 35,18: VTAB 3: PRINT
    D$;"CATALOGD";D: POKE 34,20
2030  HTAB 2: VTAB 20: PRINT "LOA
    D DATA FILE...": HTAB 2: VTAB
    22: PRINT "FILENAME:";LL$: VTAB
    22: HTAB 10: INPUT ":";FL$: IF
    FL$ = "" THEN  PRINT B$: RETURN

2040  POKE 34,23: PRINT D$;"OPEN"
    ;FL$: PRINT D$;"READ";FL$: INPUT
    N: FOR I = 1 TO N: INPUT A$(
    I): INPUT AR$(I): INPUT T(I)
    : INPUT C(I): INPUT F(I): FOR
    J = 1 TO C(I): INPUT S$(I,J)
    : NEXT J,I: PRINT D$;"CLOSE"
    : RETURN
2050  REM  SAVE ***
2060  POKE 34,4: HOME : VTAB 19: PRINT
    LN$: POKE 35,18: VTAB 3: PRINT
    D$;"CATALOGD";D: POKE 34,20
2070  HTAB 2: VTAB 20: PRINT "SAV
    E DATA FILE...": HTAB 2: VTAB
    22: PRINT "FILENAME:";LL$: VTAB
    22: HTAB 10: INPUT ":";FL$: IF
    FL$ = "" THEN  PRINT B$: RETURN

2080  POKE 34,23: PRINT D$;"OPEN"
    ;FL$: PRINT D$;"WRITE";FL$: PRINT
    N: FOR I = 1 TO N: PRINT A$(
    I): PRINT AR$(I): PRINT T(I)
    : PRINT C(I): PRINT F(I): FOR
    J = 1 TO C(I): PRINT S$(I,J)
    : NEXT J,I: PRINT D$;"CLOSE"
    : RETURN
2090  REM  DELETE ***
2100  POKE 34,4: HOME : VTAB 19: PRINT
    LN$: POKE 35,18: VTAB 3: PRINT
    D$;"CATALOGD";D: POKE 34,20
2110  HTAB 2: VTAB 20: INVERSE : PRINT
    "DELETE";: NORMAL : PRINT "
    RECORD FILE...": HTAB 2: VTAB
    22: PRINT "FILENAME:";LL$: VTAB
    22: HTAB 10: INPUT ":";FL$: IF
    FL$ = "" THEN  PRINT B$: RETURN

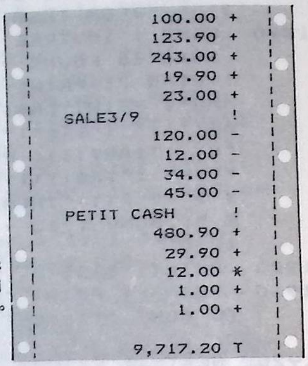
2120  POKE 34,23: PRINT D$;"DELET
    E";FL$: RETURN
2130  REM  CATALOG TO PRINTER ***

2140  POKE 34,3: HOME : VTAB 19: PRINT
    LN$: VTAB 20: HTAB 2: PRINT
    "DUMP DISK TO PRINTER...": POKE
    35,18: PRINT D$;"PR#";PS: VTAB
    3: PRINT D$;"CATALOGD";D: RETURN

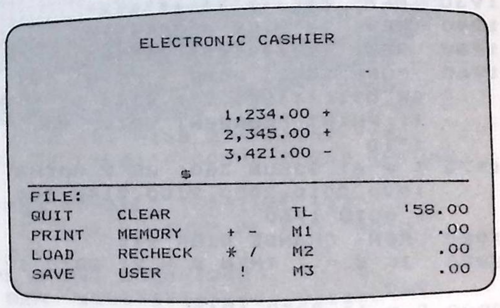
2150  REM  *********
2160  REM  ** END **
2170  REM  *********
2180  HTAB 1: VTAB 20: IF N > 0 THEN
    PRINT " THERE IS A RECORD I
    N MEMORY..": VTAB 22: PRINT
    TAB( 2);"WANT TO QUIT NOW ?
    (  ES/ O ) ": INVERSE : VTAB
    22: HTAB 23: PRINT "Y";: HTAB
    27: PRINT "N";: NORMAL : HTAB
    32: GET A$: IF A$ <  > "Y" THEN
    340
2185  TEXT : HOME : END
2190  REM  SUBROUTINES ***
2200  FOR P = 1 TO 1500: NEXT : RETURN
    : REM  DELAY ***
2210  REM  ERROR HANDLING ***
2220 EL =  PEEK (218) +  PEEK (21
    9) * 256:ER =  PEEK (222): PRINT
    D$;"PR#0": NORMAL
2230  IF ER = 255 THEN 340
2240  IF ER = 11 THEN  VTAB 22: PRINT
    B$;" WRONG INPUT"; TAB( 39):
    GOTO 2310
2250  IF ER = 5 THEN  PRINT D$;"D
    ELETE";FL$: VTAB 22: PRINT B
    $;" FILE NOT FOUND"; TAB( 39
    ): GOTO 2310
2260  IF ER = 4 THEN  HTAB 1: VTAB
    22: PRINT B$;" DISK WRITE PR
    OTECTED"; TAB( 39): GOTO 231
    0
2265  IF ER = 53 THEN  HOME : VTAB
    10: PRINT "RECORD NO. EXCEED
    S RECORD VOLUME (N>VN)": VTAB
    12: PRINT "SAVE CURRENT DATA
    AND QUIT": VTAB 14: PRINT "
    THEN CHANGE LINE ";: INVERSE
    : PRINT "130  VN=";: NORMAL
    : HTAB 21: PRINT "  ";: HTAB
    27: PRINT "[GREATER VAL]": GOTO
    2270
2270  IF ER = 254 THEN  HTAB 1: VTAB
    22: PRINT B$;" ILLEGAL INPUT
    "; TAB( 39): GOTO 2310
2280  IF ER = 9 THEN  HTAB 1: VTAB
    22: PRINT B$;" DISK FULL,SAV
    E CANCELLED"; TAB( 39): PRINT
    D$;"DELETE";FL$: GOTO 2310
2290  IF ER = 8 THEN  VTAB 22: HTAB
    1: PRINT B$;" I/O ERROR"; TAB(
    39): GOTO 2310
2300  TEXT : HOME : NORMAL : PRINT
    B$;"ERROR CODE ";ER;" AT LIN
    E ";EL: PRINT "CONSULT YOUR
    DOS OR APPLOESOFT MANUAL": END

2310  GOSUB 2200: GOTO 340
```

萬能電子計算機



圖(2) 利用萬能電子計算機打印出來的PRINTOUT例子。大家可在上加上註釋欄。



圖(1) 程式執行時之畫面狀態。

假如閣下日常要作大量繁複的計算，定必感到那是件十分考耐性的工作。因爲就算有計算機的協助，亦會間中按錯鍵，以致要重覆計算多次才能得出正確答案，徒然浪費不少的時間。

本程式基本上模擬了一具有多項功能的計算機；如果你有印字機的話，本程式更是一具能出紙的計算機。這具計算機完全發揮了電腦龐大記憶的特點，因爲每筆計算都儲在記憶內，你便可以隨意地修改、加入、刪除及觀看每筆計算。除此之外，整個計算都可存在磁碟中，以便日後有需要的時候取用。

使用本程式

由於整個程式都以BASIC寫成，各位照着列表鍵入後，以SAVE ELECTRONIC CASHIER 存放在磁碟後便可以RUN。

程式執行了後便可看見一個如圖一的畫面。左下方是本程式其他功能的選擇，而右下方則顯示了總和（TL）及9個記憶中其中三個（M1－M3）的值，以方便計算之時有所參考。在畫面中央"$"符號的旁邊可見到一個閃動的游標，這裏你可以直接打入每筆的計算，又或者選擇本程式的其他功能。不過你的輸入只能是某功能的字母或是一個數值加上一個數學符號（＋，－，＊或／），而數值的範圍由－10 000,000至100,000,000，太大的數值或輸入中有"："或"，"符號都會導致錯誤。這點大家必須緊記。

數學功能

本程式所提供的運算功能有加（＋），減（－），乘（＊）及除（／）。除了＊以外，其他符號輸入之時按不按SHIFT也可以，即；與＋相通，？與／相通等。

就如一般的計算機一樣，輸入一條算式的次序就是先打進一個數值再跟着一個運算子。例如你打入　5　＋<RETURN>，在TL（總和）一欄便可看到5・00。再按多一個　＋<RETURN>，TL就會顯示10・00，因爲5加5等於10。其他的－，＊及／都是使用類似的方式輸入，各位多加練習便會明白。有一個特別的符號"！"可以在有需要的時候使用。如單單打進！<RETURN>，那麼TL的數值就會在左面出現，以作註解或SUB－TOTAL之用。另外若在！之前加上什麽（包括文字）都會出現在左面，以作註解之用。這項功能能讓你更清楚各項的收支。

其他功能

<Q>　QUIT是離開此程式。

<P>　PRINT 是把整份計算列印於印字機以作保存之用，記緊要把印字機開着。假如印字機控制咭不是插在SLOT 1 可更改行5040。

<L>和<S>　LOAD和SAVE　用作磁碟的存取。這項功能方便了日後對是次計算的修訂。你需要使用一個檔名，如只輸入一個？就得出磁碟的CATALOG。

<C>　CLEAR 是把一切的計算紀錄清除，但不會清除M1－M9 各個記憶。

<M>　MEMORY 因爲這是電腦的關係，故記憶系統是較一般的計算機完善。打入M　＃　<RETURN>，你便能選擇需要多少個記憶（程式起始之時只有三個記憶M1－M3）。設定一個記憶，先打進M及記憶的號碼（例如M1），看到了左面出現了反相的M1後在$裏輸入一數值及＋　<RETURN>便可。清除一記憶，可把其設定爲0。記憶的數值可隨時取用，用以取代要重複輸入的數值。

<R>　RECHECK　使你能將整份計算的任何一筆作出修改。按N或L鍵可使記錄作相應的上及下捲動，把需要更改的紀錄移至最底再按R鍵便能把其更改，方法就是輸入一數值及一個＋符號。更改了後按E鍵便離開此狀態。

<U>　是保留給USER　用家自定功能。各位如有需要，可在此處加進自己喜愛的功能，例如加上一些科學函數的處理等來迎合自己的需要。唯一要做的是在行500中的IF…THEN 敍述加上其代表字符。

總括來說，本程式非常容易使用及有效地簡化了繁重的計算工作。就算與一些專業軟件如VISICALC　等相比，也是不遑多讓的，實在是辦公室及收銀員的好帮手。■

```
10  REM  **********************
20  REM  * ELECTRONIC CASHIER *
30  REM  *        BY          *
40  REM  *     COMPUHOUSE     *
50  REM  **********************
60  GOTO 5010
70  REM
FIND POSITION

80 J = LR + B
90  REM
POSITION (CONT.)

100 J = J - B: IF J = E THEN  RETURN

110  IF  MID$ (R$,J,B) = T$ THEN
    RETURN
120  GOTO 100
130  REM
ELIMINATE SPACES

140 T$ = " ":J = LR + B
150  GOSUB 100: IF  NOT J THEN  RETURN

160  IF J = LR THEN LR =  LR - B:R
    $ =  LEFT$ (R$,LR): GOTO 150

170  IF J = B THEN LR = LR - B:R$
     =  RIGHT$ (R$,LR: RETURN
180 R$ =  LEFT$ (R$,J - B) +  RIGHT$
    (R$,LR - J):LR = LR - B: GOTO
    150
190  REM
NON # CHECK

200 J = LR + B
210 J = J - B: IF  NOT J THEN  RETURN

220 T$ =  MID$ (R$,J,B): IF J = B
    AND T$ = "-" THEN 210
230  IF T$ <  > "/" AND T$ > "*" AND
    T$ < ":" THEN 210
240  RETURN
250  REM
CHECK DECIMAL PLACES

260 K = B: IF  NOT  VAL (R$) THEN
    R$ = ".00":LR = D:K = E: RETURN

270 LR =  LEN (R$): IF LR > D THEN
    IF  MID$ (R$,LR - D,B) = "E
    " THEN K = E: RETURN
280 T$ = ".": GOSUB 80: IF  NOT J
    THEN R$ = R$ + ".00":LR = L
    R + D: GOTO 320
290  IF LR - J > B THEN J = J - B
    : RETURN
300  IF LR - J THEN R$ = R$ + "0"
    :LR = LR + B: GOTO 320
310 R$ = R$ + "00":LR = LR + C
320 J = LR - D: RETURN
330  REM
REFORMAT

340  GOSUB 260: IF  NOT K THEN  RETURN

350  IF  LEFT$ (R$,B) = "-" THEN
    K = C
360 J = J - D: IF J < K THEN  RETURN

370 R$ =  LEFT$ (R$,J) + "," +  RIGHT$
    (R$,LR - J):LR = LR + B: GOTO
    360
380  REM
INPUT ERROR

390  RESTORE : FOR K = B TO J: READ
    T$: NEXT : VTAB VV: HTAB 28:
     INVERSE : PRINT T$;G$;: NORMAL
    : GOTO 430
400  REM
MAIN INPUT

410  VTAB VV: HTAB 15: PRINT NN$;

420  REM
INPUT (CONT.)

430  HTAB 15: INPUT "$";R$: VTAB
    VV: HTAB 27: INVERSE : PRINT
    IN$: NORMAL :LR =  LEN (R$):
     IF  NOT LR THEN J = C: GOTO
    390
440  IF  RIGHT$ (R$,B) = "!" THEN
    C$ = "!": RETURN
450  GOSUB 140: IF  NOT LR THEN J
     = C: GOTO 390
460 C$ =  RIGHT$ (R$,B)
470  IF LR > 11 THEN J = D: GOTO
    390
480  RETURN
490  REM
MATH FUNCTIONS

500 J = E: IF C$ = "-" OR C$ = "=
    " THEN C$ = "-":S = S - R: RETURN

510  IF C$ = "+" OR C$ = ";" THEN
    C$ = "+":S = S + R: RETURN
520  IF (C$ = "/" OR C$ = "?") AND
    R <  > E THEN C$ = "/":S = S
    / R: RETURN
530  IF C$ = "*" THEN C$ = "*":S =
    S * R: RETURN
540  REM
FOR EXPANSION FUNCTIONS

550 J = C: RETURN : REM  IF J TH
    EN ERROR
560  REM
SCROLL WINDOW UP

570  VTAB VV - B: POKE 35,VV - B:
    PRINT : RETURN
580  REM
PRINT NEXT

590  GOSUB 570: HTAB 13
```

```
600  REM
PRINT NEXT (CONT.)

610 EN = 14: GOSUB 670: PRINT " "
    ;C$;: TEXT : RETURN
620  REM
PRINT TOTAL

630 M = E
640  REM
PRINT MEMORY

650  VTAB VV + C + M:R$ =   STR$ (
     S): HTAB 27:EN = 14
660  REM
PRINT ONE

670  GOSUB 340: IF LR = 14 THEN 7
     00
680  IF LR > 14 THEN  GOSUB 720: IF
     LR = 14 THEN 700
690  FOR J = B TO EN - LR: PRINT
     " ";: NEXT
700  PRINT R$;: RETURN
710  REM
TOO LONG

720 T$ = ".": GOSUB 80:LR = 14: IF
    J = 14 THEN LR = 13
730 R$ =  LEFT$ (R$,LR): RETURN
740  REM
PREPARE #

750  GOSUB 200: IF J THEN J = B: RETL
760 R =  VAL (R$): IF R =  > R1 OR
    R =  < R2 THEN J = D: RETURN

770 J = E: RETURN
780  REM
PRINT NEW SCREEN
790 V = E:II = I - VV + C: IF II <
    B THEN II = B:V = VV - I - B

800  FOR II = II TO I:V = V + B: VTAI
     V: POKE 36,E:R$ = R$(II):C$ =
     C$(II): HTAB 13: IF C$(II) =
     "!" THEN  GOSUB 1960: NEXT :
     RETURN
810  GOSUB 610: NEXT : RETURN
820  REM
FIND MEM #

830  IF  NOT J THEN J = 4: RETURN

840 M =  VAL ( MID$ (R$,J,B)): IF
    M < B OR M > MN THEN J = 4: RETL
850 J = E: RETURN
860  REM
GET C$

870 LR = LR - B:R$ =  LEFT$ (R$,L
    R): RETURN
880  REM
PRINT MD$

890 H = 14
900  VTAB VV: HTAB H: INVERSE : PRINT
     MD$: NORMAL : RETURN
1000  REM
ERROR

1010  GOSUB 390: GOTO 1050
1020  REM
START

1030  GOSUB 410
1040  REM
MODE CHECK

1050 TT$ =  LEFT$ (R$,B): IF LR =
     B AND TT$ = "!" THEN R$ =  STR$
     (SUM): GOSUB 260: GOTO 1930
1055  IF C$ = "!" THEN 1200
1060  IF TT$ = "M" THEN 1280
1070  IF TT$ = "Q" THEN 2300
1080  IF  NOT I THEN  IF TT$ = "P
     " OR TT$ = "S" OR TT$ = "R" THEN
     J = 7: GOTO 1010
1090  IF TT$ = "P" THEN 2000
1100  IF TT$ = "L" THEN TT$ = "LO
     AD": GOTO 2060
1110  IF TT$ = "S" THEN TT$ = "SA
     VE": GOTO 2060
1120  IF TT$ = "R" THEN 1550
1130  IF TT$ = "C" THEN 1900
1140  IF TT$ = "U" THEN 2330
1150  IF LR = B THEN  IF I AND C$
     (I) <  > "!" AND (TT$ = "+" OR
     TT$ = ";" OR TT$ = "-" OR TT
     $ = "=" OR TT$ = "*" OR TT$ =
     "/" OR TT$ = "?") THEN C$ =
     TT$:R$ = R$(I): GOTO 1240
1170  REM
THIS AREA FOR EXPANSION
     MODES

1200  IF LR = B THEN J = C: GOTO
     1010
1210  GOSUB 870: IF C$ = "!" THEN
     1930
1220  GOSUB 750: IF J THEN 1010
1230  REM
REENTRY

1240  IF I = TR THEN J = 6: GOTO
     1010
1250 S = SUM: GOSUB 500: IF J THEN
     1010
1260 I = I + B:R$(I) = R$:C$(I) =
     C$: GOSUB 630:SUM = S:R$ = R
     $(I): GOSUB 590: GOTO 1030
1270  REM
MEMMORY

1280 T$ = "T": GOSUB 80: IF J THEN
     1460
```

```
1290 T$ = "C": GOSUB 80: IF J THEN
     1430
1300  IF  RIGHT$ (R$,B) = "#" THEN
     1510
1310 J = LR: GOSUB 830: IF  NOT J
     THEN 1360
1320 J = LR - B: GOSUB 830: IF J THEN
     1010
1330  REM
MEMORY TO TOTAL

1340  GOSUB 870:R = M(M):R$ =  STR$
     (R): GOTO 1240
1350  REM
DIRECT TO MEMORY

1360 MD$ = "M" +  STR$ (M):H = 13
     : GOSUB 900: GOSUB 410: GOTO
     1380
1370  GOSUB 390: REM   MEMORY ERR
     OR INPUT
1380  IF LR = B THEN J = 5: GOTO
     1370
1390  GOSUB 870: GOSUB 750: IF J THEN
     1370
1400 S = M(M): GOSUB 500: IF J THEN
     1370
1410  GOSUB 650:M(M) = S:MD$ = "
     ": GOSUB 900: GOTO 1030
1420  REM
CLEAR MEMORY

1430 J = J - B: GOSUB 830: IF J THEN
     1010
1440 M(M) = E:S = E: GOSUB 650: GOTO
     1030
1450  REM
TOTAL TO MEMORY

1460 C$ =  MID$ (R$,J - B,B):J =
     J - C: IF J < B THEN J = 5: GOTO
     1010
1470  GOSUB 830: IF J THEN 1010
1480 S = M(M):R = SUM: GOSUB 500:
     IF J THEN 1010
1490  GOSUB 650:M(M) = S: GOTO 10
     30
1500  REM
CHANGE # OF MEMORIES

1510  HOME : PRINT "HOW MANY MEMO
     RIES DO YOU NEED?";F$;"CHOOS
     E FROM 1 TO 9:  ";: POKE  -
     16368,E: GET T$: IF T$ < "1"
     OR T$ > "9" THEN  PRINT G$:
     GOTO 1510
1520 MN =  VAL (T$): HOME : GOSUB
     5210: GOSUB 2270:S = SUM: GOSUB
     630: GOSUB 2220: IF I THEN  GOSUB
     790
1530  GOTO 1030
1540  REM
RECHECK

1550  VTAB VV + C: HTAB 27: CALL
     - 868: CALL SS:MD$ = "R": POKE
     36,E: INVERSE : PRINT "E";F$
     ;"R";F$;"I";F$;"N";F$;"L";: NORMAL
     : VTAB VV + C: POKE 32,B
1560  PRINT "XIT"; SPC( 9);F$;"EV
     ISE"; SPC( 8);F$;"NSERT" SPC(
     10);F$;"EXT" SPC( 8);F$;"AST
     ";:II = E: TEXT : POKE 35,VV
     - B: HOME : TEXT :RV = E
1570  REM
NEXT
1580  IF II = I THEN 1690
1590 II = II + B:R$ = R$(II):C$ =
     C$(II): IF C$ = "!" THEN  GOSUB
     570: HTAB 13: GOSUB 1960: TEXT
     : GOTO 1620
1600  GOSUB 590
1610  REM
GET RECHECK INPUT

1620  POKE 36,E: GOSUB 890: VTAB
     VV: HTAB 15: PRINT NN$
1630  VTAB VV: HTAB 15: POKE  - 1
     6368,E: GET TT$: IF TT$ = "E
     " THEN 1870
1640  IF TT$ = "R" THEN 1760
1650  IF TT$ = "N" THEN 1580
1660  IF TT$ = "L" THEN 1710
1670  IF I = TR THEN 1690
1680  IF TT$ = "I" THEN 1850
1690  PRINT G$;: GOTO 1630
1700  REM
LAST

1710  IF II < C THEN 1690
1720 II = II - B:K = II - VV + C:
     CALL SD: IF K < B THEN  GOTO
     1620
1730 C$ = C$(K):R$ = R$(K): IF C$
     = "!" THEN  VTAB B: HTAB 13
     : GOSUB 1960: GOTO 1620
1740 EN = 24: HTAB D: GOSUB 670: PRINT
     " ";C$;: GOTO 1620
1750  REM
REVISE

1760  VTAB VV: HTAB 16: PRINT R$(
     II);C$(II);
1770  GOSUB 430: GOTO 1790
1780  GOSUB 390: REM   REVISE ERR
     OR INPUT
1790  IF LR < C THEN J = 5: GOTO
     1780
1800 RV = B: GOSUB 870:R$(II) = R
     $: VTAB VV - B: HTAB 13: IF
     C$ = "!" THEN C$(II) = "!": GOSUB
     1960: GOTO 1620
1810  GOSUB 750: IF J THEN 1780
1820  GOSUB 500: IF J THEN 1780
1830  GOSUB 610:C$(II) = C$: GOTO
     1620
1840  REM
INSERT
```

```
1850 I = I + B:II = II + B: GOSUB
   570: TEXT : FOR K = I TO II STEP
   - 1:R$(K + B) = R$(K):C$(K +
   B) = C$(K): NEXT :MD$ = "I":
   GOSUB 890:MD$ = "R": GOSUB
   410: GOTO 1790
1860 REM
EXIT
1870 MD$ = " ":S = SUM: CALL SR: IF
   RV THEN S = E: FOR II = B TO
   I:C$ = C$(II):R =  VAL (R$(I
   I)): GOSUB 500: NEXT : GOSUB
   790:SUM = S
1880  GOSUB 630: GOTO 1030
1890  REM
CLEAR
1900 P$ = " CLEAR (Y/N)": GOSUB 1
   980: IF T$ <  > "Y" THEN 103
   0
1910 I = E:SUM = E: POKE 35,VV -
   B: HOME : TEXT :S = E: GOSUB
   630:FL$ = "": GOSUB 2270: GOTO
   1030
1920  REM
COMMENTS
1930  GOSUB 570: HTAB 13: IF TT$ =
   "M" THEN J = LR: GOSUB 830: IF
   NOT J THEN R$ =  STR$ (M(M
   ): GOSUB 260
1940  GOSUB 1960: TEXT :I = I + B
   :C$(I) = "!":R$(I) = R$: GOTO
   1030
1950  REM
PRINT COMMENTS
1960  PRINT R$;:LR =  LEN (R$): FOR
   J = B TO 15 - LR: PRINT " ";
   : NEXT : PRINT "!";: RETURN
1970  REM
PRINT P$
1980  VTAB VV: HTAB 15: FLASH : PRINT
   P$;G$;: NORMAL : POKE - 163
   68,E: GET T$: RETURN
1990  REM
TO PRINTER
2000  CALL SS: POKE 35,VV - B: HOME
   : PRINT "TURN ON PRINTER.  C
   ONTINUE?  (Y/N)  ";: POKE -
   16368,E: GET T$: IF T$ <  >
   "Y" THEN 2040
2010  POKE 36,E: CALL - 868:EN =
   14: POKE 32,12: POKE 33,28: VTAB
   VV - B: PR# P: FOR II = B TO
   I:R$ = R$(II):C$ = C$(II): PRINT
   : IF C$ = "!" THEN  GOSUB 19
   60: GOTO 2030
2020  GOSUB 670: PRINT " ";C$;
2030  NEXT : PRINT F$:R$ =  STR$
   (SUM): GOSUB 670: PRINT " T"
   ;F$;F$
2040  PR# E: TEXT : CALL SR: GOTO
   1030
2050  REM
SAVE/LOAD
2060  CALL SS:FB$ = FL$
2070  GOSUB 2280: VTAB VV + B: HTAB
   38: INVERSE : PRINT "   ": VTAB
   VV + B: PRINT TT$;":  ";: NORMAL
   : INPUT "";T$:T =  LEN (T$):
   IF  NOT T THEN  CALL SR: GOTO
   2200
2080 E$ =  LEFT$ (T$,B): IF E$ =
   " " AND T - B THEN T = T - B
   :E$ =  RIGHT$ (E$,T): GOTO 2
   080
2090  IF E$ = " " THEN  PRINT G$:
   GOTO 2250
2100  IF E$ = "?" AND T = B THEN
   2240
2110  IF T > 30 THEN T = 30
2120 FL$ =  LEFT$ (T$,T): VTAB VV
   - B: PRINT D$: PRINT D$;"OP
   EN";FL$: IF TT$ = "LOAD" THEN
   2160
2130  REM
SAVE
2140  PRINT D$;"WRITE";FL$: PRINT
   I: PRINT MN: FOR II = B TO I
   : PRINT R$(II): PRINT C$(II)
   : NEXT : PRINT SUM: FOR II =
   B TO MN: PRINT M(II): NEXT :
   PRINT D$;"CLOSE": CALL SR: GOTO
   2200
2150  REM
LOAD
2160  PRINT D$;"READ";FL$: INPUT
   I: INPUT MM: FOR II = B TO I
   : INPUT R$(II): INPUT C$(II)
   : NEXT : INPUT SUM: FOR II =
   B TO MM: INPUT MM(II): NEXT
   : PRINT D$;"CLOSE": CALL SR:
   GOSUB 2270:S = SUM: GOSUB 6
   30: POKE 35,VV - B: HOME
2170  IF MM > MN THEN MM = MN
2180  PRINT "DO YOU WISH TO REPLA
   CE THE CURRENT M1 TOM";MM;"
   WITH THOSE SAVED WHEN THIS F
   ILE WAS  SAVED?  (Y/N) ";G$;
   : GET T$: HOME : TEXT : IF T
   $ = "Y" THEN  FOR M = B TO M
   M:M(M) = MM(M): NEXT : GOSUB
   2220
2190  GOSUB 790
2200  GOSUB 2270: GOTO 1030
2210  REM
PRINT NEW MEMORIES
2220  FOR M = B TO MN:S = M(M): GOSUB
   650: NEXT : RETURN
```

```
2230  REM
CATALOG

2240  POKE 35,VV - B: HOME : PRINT
   D$;"CATALOG": TEXT
2250  GOSUB 2270: GOTO 2070
2260  REM
PRINT FL$

2270  INVERSE
2280  HTAB B: VTAB VV + B: PRINT
   "FILE:  ";FL$; SPC( 33 -  LEN
   (FL$)): NORMAL : RETURN
2290  REM
QUIT

2300 P$ = " QUIT? (Y/N)": GOSUB 1
   980: IF T$ <  > "Y" THEN 103
   0
2310  TEXT : END
2320  REM
USER

2330  PRINT G$;: GOTO 1030
2340  REM   FOR USER EXPANSION
5000  REM
INITIALIZATION

5010  HIMEM: 37375: REM    DON'T C
   HANGE
5020 TR = 200: REM   INPUT RECORD
   LIMIT MAY BE CHANGED
5030 VP = 6: REM   MINIMUM RESERV
   ED LINES FOR LOWER INFORMATI
   ON WINDOW - MUST NOT BE LESS
    THAN 6
5040 P = 1: REM   PRINTER SLOT #.
    IF NO PRINTER EXISTS MAKE
   P = 0
5050  TEXT : HOME : HTAB 12: INVERSE
   : PRINT "ELECTRONIC CASHIER"
   : NORMAL : ONERR  GOTO 5310
5060  DIM R$(TR + 1),C$(TR + 1):E
   = 0:B = 1:C = 2:D = 3:MN =
   VP - D:R1 = 100000000:R2 = -
   10000000:MD$ = " ":F$ =  CHR$
   (13):G$ =  CHR$ (7):D$ =  CHR$
   (4):B$ =  CHR$ (8):FL$ = "":
   IN$ = "              ":NN$ =
   "    "
5070  DATA   "NON NUMBER","NO COM
   MAND","TOO LARGE","WHICH MEM
   ORY","UNDEFINED","MEMORY FUL
   L","MEMORY EMPTY"
5080  DATA    162,0,189,0,4,157,0
   ,146,189,0,5,157,0,147,189,0
   ,6,157,0,148,189,0,7,157,0,1
   49,232,208,229,96,162,0,189,
   0,149,157
5090  DATA    0,7,189,0,148,157,0
   ,6,189,0,147,157,0,5,189,0,1
   46,157,0,4,232,208,229,96,16
   2,18,189,125,3,133,252,189,1
   07,3,133,253
5100  DATA    202,240,21,189,125,
   3,133,254,189,107,3,133,255,
   160,39,177,254,145,252,136,1
   6,249,48,222,169,0,133,36,13
   3,37,32,34,252,76,156,252
5110  DATA    4,4,5,5,6,6,7,7,4,4
   ,5,5,6,6,7,7,4,4,0,128,0,128
   ,0,128,0,128,40,168,40,168,4
   0,168,40,168,80,208
5120  FOR J = 1 TO 7: READ T$: NEXT
   : FOR J = 768 TO 911: READ T
   : POKE J,T: NEXT :SD = 828:S
   V = 829:SS = 768:SR = 798: GOSUB
   5210: GOSUB 630: GOSUB 2220:
    GOTO 1030
5200  REM
WARMSTART

5210 VV = 24 - VP: IF VV + MN > 2
   1 THEN VV = 21 - MN
5220  POKE SV,VV - B: VTAB VV: INVERSE
   : PRINT IN$;: HTAB 27: PRINT
   IN$;"FILE:"; SPC( 35);"Q";F$
   ;"P";F$;"L";F$;"S": VTAB VV +
   C: HTAB 9: POKE 32,8: PRINT
   "C";F$;"M";F$;"R";F$;"U"
5230  VTAB VV + C: HTAB 17: POKE
   32,24: PRINT "TL";: FOR J =
   B TO MN: PRINT F$;"M";J;: NEXT
   : TEXT
5240  VTAB VV + D: HTAB 19: PRINT
   "+";: HTAB 21: PRINT "-": HTAB
   19: PRINT "*";: HTAB 21: PRINT
   "/": HTAB 20: PRINT "!": NORMAL

5250  VTAB VV + C: HTAB C: POKE 3
   2,B: PRINT "UIT";F$;"RINT";F
   $;"OAD";F$;"AVE": VTAB VV +
   C: HTAB 9: POKE 32,9: PRINT
   "LEAR";F$;"EMORY";F$;"ECHECK
   ";F$;"SER": TEXT : RETURN
5300  REM
ONERR GOTO

5310  TEXT :X =  PEEK (222):Y =  PEEK
   (218) +  PEEK (219) * 256: IF
   X = 16 THEN  PRINT F$;"SYNTA
   X ERROR IN LINE ";Y;G$: END

5320  IF X = 254 AND Y = 2160 THEN
   PRINT D$: PRINT D$;"CLOSE":
   GOTO 5350: REM    Y = LINE #
   WHICH READS NEW FILE IN LOA
   D MODE
5330  IF X < 4 OR X > 13 THEN  PRINT
   F$;"ERROR = ";X;" IN LINE ";
   Y;G$: END
5340  IF X = 5 THEN  VTAB VV - 2:
   PRINT D$: PRINT D$;"DELETE"
   ;FL$
5350  VTAB VV + 1: INVERSE : PRINT
   "I/O ERR";G$;: NORMAL : GET
   T$:FL$ = FB$: CALL SR: IF X =
   254 THEN 1910
5360  GOTO 1030
```

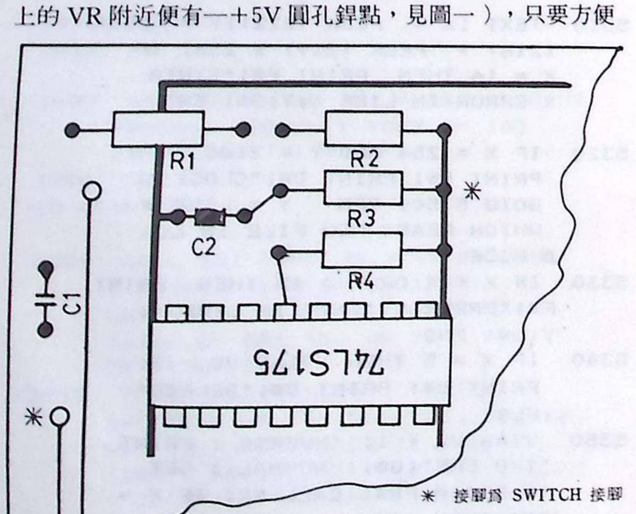
非常輕易的改良：

16K WILDCARD

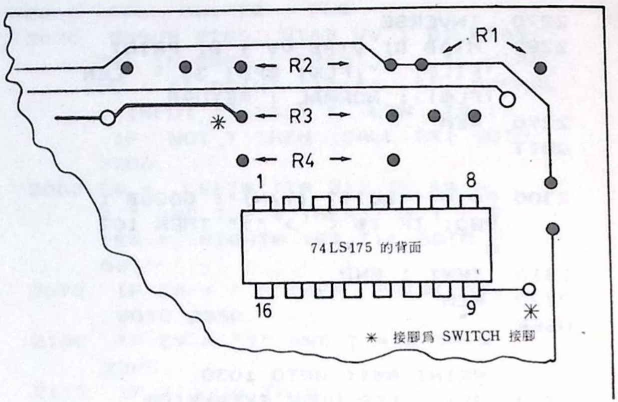
—朱志誠—

「硬」件，顧名思義，是硬的，死板的；而「HARD-SWITCH」，也是死板的，「軟」件不能控制它。利用這點，便能破解一些受保護的軟件，當然除了有「HARD-SWITCH」外，還要有軟件的幫助，才能達到目的。基本原理就是要靠預先將一些特別資料放在一張16K RAM-CARD上，用「HARDSWITCH」將它鎖住，然後才RUN受保護程式。程式進行中的某個時間按下RESET，控制權便落在RAMCARD上，只要RAMCARD能控制電腦，受保護程式便沒法躲避了。

爲免大家不知筆者說甚麼「HARDSWITCH」，還是先介紹硬件部份的改良。一張普通的16K RAMCARD上有一粒74LS175，用途是將＄C080至＄C08F解碼爲一些「SOFTSWITCH」，用來控制READ ENABLE，WRITE ENABLE及其他。其實74LS175的四個輸入，只是供給三個位址線（A0，A1及A3）輸入，即從此四輸入線中只能測出現在位址的最低位（十六進位）的正確數字，即＄xxx0至＄xxxF（x表示未知數）。其餘的三個數字（十六進位）是從DEVICE SELECT線取得。當DEV SEL線電平降低再上升（即正邊緣）時，74LS175才知道位址爲＄C08x（當RAMCARD插在SLOT0上時）。所以如果將DEV SEL永遠接着+5V，則電平永遠是高，而RAMCARD便會以爲程式沒有使用到位址＄C08x，RAMCARD不能開，也不能被寫入資料。所以硬件的改良只要將一個最普通的單刀單擲開關接在+5V和DEV SEL之間便行。讀者請先行找出RAMCARD上的74LS175第9脚，將電線銲在此接脚上以印刷線路連接着的小圓孔銲點（如圖一所指的銲點）。而+5V可從CARD上的很多位置找到（例如在CARD上的VR附近便有一+5V圓孔銲點，見圖一），只要方便



16K RAMCARD 的正面



圖(1)　16K RAMCARD 的背面

銲接便可，一張「16K WILDCARD」就這樣產生了（以下稱它爲「16K WILDCARD」。接好線後，最好檢查一次。插上16K WILDCARD於SLOT0後，就可開始說明用法了。

首先，關了SWITCH才開機，目的使電腦能打開16K WILDCARD，將資料寫進去，跟着按

```
]CALL-151
*D000<D000.FFFFM
*FF6A:A9
*C083
```

啊！PROMPT變成"）"了！如果你現在按

）FFFC：59FF

那麼按RESET後便跳到舊的RESET控制程式，它不會BOOT碟，亦不會利用位址＄3F2至＄3F4，不會被受保護程式控制。所以這是一個很好用的RESET控制程式。按下

＊C080

就進入了READ RAM和WRITE PROTECT MODE，然後再將SWITCH開了（即CLOSE CIRCUIT），那麼你按C081，C082或其他鍵都不能改變現時的READ RAM ONLY MODE，因此便成爲了一個RAMCARD WRITE PROTECT，其他程式只會以爲這些RAMCARD上的資料是ROM內的，只能READ出來，不能WRITE進去。當然，在軟件方面只作以上的改動是不夠的。列表一是一個更加有用的PROGARM，它是供給一張插在SLOT0的「16K WILDCARD」用的。打入程式列表後，按

＊BSAVE RAM-WILD·SOFT，A＄C00，L＄89

存入磁碟後，按C00G，程式會用「畢」聲來提醒你關SWITCH，然後自動將ROM搬到RAMCARD去，更改RESET控制程式（注意當程式運行中不要RESET）。以後每當按過RESET後，電腦便會要求按鍵，按ESC鍵便

非常輕易的改良：16K WILD CARD

```
******************
* RAM-WILD.SOFT *
*   A$C00,L$89    *
******************
0C00- AD 83 C0 AD 83 C0 AD 30
0C08- C0 A0 00 88 D0 FD AD 00
0C10- D0 EE 00 D0 CD 00 D0 F0
0C18- E7 AD 81 C0 A0 00 A9 00
0C20- 85 42 85 3C A9 FF 85 3E
0C28- 85 3F A9 D0 85 43 85 3D
0C30- 20 2C FE A0 46 B9 42 0C
0C38- 99 62 FA 88 10 F7 AD 80
0C40- C0 60 8D 10 C0 AD 00 C0
0C48- 10 FB 8D 10 C0 C9 9B F0
0C50- 32 AD 8B C0 AD 8B C0 AD
0C58- 30 C0 A0 00 88 D0 FD AD
0C60- 00 D0 EE 00 D0 CD 00 D0
0C68- F0 E7 A0 00 A9 00 85 3C
0C70- 85 42 85 3D A9 D0 85 43
0C78- A9 FF 85 3E A9 0F 85 3F
0C80- 20 2C FE AD 80 C0 4C 59
0C88- FF
```

列表一

會執行舊式重定循環；如果按其他鍵就會用「畢」聲來叫你關 SWITCH，使電腦能將＄0-＄FFF 的資料放在＄D000-＄DFFF 內（RAMCARD 內），跟着便做一個舊式重定循環，和新式重定遁環不同的，就是它不會 BOOT 碟，也不會用到位址＄3F2至＄3F4，目的是保護位址＄0至＄FFF 內的程式或資料，現時＄D000-＄DFFF 是存着 APPLESOFT INTERPRETER 程式的＄D000-＄DFFF 部份，如果要＄0-＄FFF 的資料，可按＄C088。以上只是 PROGRAM 一例，讀者可自行寫一個類似眞正 WILDCARD MENU 的控制程式。但可惜一點，就是筆者暫時還未完成研究在 RESET 後將 ADDRESS BUS 存放在 RAMCARD 上。研究成功後，便可將一個暫停了的程式繼續運行，情形有如眞正 WILDCARD 一般。

16K WILD CARD 的缺點

雖然這張16K WILDCARD 有不少流弊，它不能 RUN 64K 的程式，而且每次開機都要先 BOOT 起16K WILDCARD 所需的 PROGRAM；但它比一張眞正的 WILDCARD 更具彈性，用家可自行創作軟件供它使用，最大的特點還是容易改裝。總之，各有各的好處，希望讀者能善用其長。

最後筆者要多謝第十六期的 BABEL 君；提供了74LS175的一些資料，更啓發了本人造出此「16K WILDCARD」的靈感。本人非常同意 BABEL 君的說法—「這雖然只是一個很少的改良，却帮了筆者不少大忙」。希望下次能介紹一個更好的 RESET 控制程式，供給這張16K WILDCARD 使用。

啊！差點兒忘記了，其實這張16K WILDCARD 還有一個特別用途，就是可作爲一個模擬 ROM，將自己改良過的 PROGARM（例如將 AUTOSTART ROM 改爲 MONITOR ROM，或將 APPLESOFT BASIC 改爲 INTEGER BASIC，又或者將 ROM 裏的 COMPUTER 字樣改爲 APPLE][等等）放在16K WILDCARD 上，然後用 SWITCH 將它保護着，那麼大家便豈不是有一張模擬的 EPROM CARD 麼！

製作過程

以下是詳細的製作步驟：

(1)先找到16K WILDCARD 上的74LS175的位置，它附近有幾個電阻。

(2)這幾個電阻 R2、R3、R4的共同端爲＋5V 接線（參看圖一），將電線接在此。

(3)反轉 RAMCARD，找出74LS175的背面和它的第9腳，從這裏有一印刷線路與 IC 平行地延出，經過一個圓孔形銲點後，到達 RAMCARD 正面，再以印刷線路連接 DEVSEL 線。所以將第二條電線接在此圓孔形銲點。

(4)將這兩條電線的另一線頭接往一個 SWITCH。這兩條電線要比較長爲佳（大約二呎），最好改用單一的屏敝線。■

如有疑問，請電3-312007 編輯部查詢

投稿規則

(一) 本刊絕對公開，歡迎任何形式的投稿，包括：軟件程式，電腦硬件 REVIEW 及製作，商用、遊戲軟件 REVIEW，技巧使用，各種電腦語言介紹詳述等等。

(二) 來稿以原稿紙單面橫寫爲合。

(三) 軟件程式投稿，如程式有一定長度，請附同磁碟或磁帶一併寄來，程式必須加以清楚說明、用法及原理。

(四) 寄來之磁碟或磁帶於稿件發表後即退回。

(五) 稿件字數長短不限。

(六) 請勿一稿兩投。

(七) 所有投稿必須附上作者中英文眞實姓名、通信地址及電話（方便電腦紀錄），發表時筆名隨意。

(八) 來稿可寄九龍中央郵局信箱 71193 號電腦時代收，信封面請註明〔投稿〕，或親身携來本刊。

ADVENTURE CONSTRUCTION SET：創作冒險遊戲大躍進

林智勤

ADVENTURE CONSTRUCTION SET (以下簡稱ACS)是ELECTRONIC ARTS之新作，它是繼MUSIC CONSTRUCTION SET 和PINEBALL CONSTRUCTION 後另一個創作佳品。

用者只要BOOT 起ACS後，出了TITLE而有音樂產生後，可按JOY STICK中的BOTTON 0去看看DEMO ，也可直接按SPACE BAR去到達MENU ，按M可以製造一隻ACS的DATA DISK，還有ACS會問你有多少部磁碟機，若只有一部便按"1"，有兩部按"2"，ACS也會問你用甚麼工具控制，按BOTTON 表示用JOY STICK ，按SPACE BAR表示用KEYBOAR—D。若ACS用RIGHT SIDE UP表示磁碟正面，UPSIDE DOWN 表示背面。按下M後，會出現另一ME—NU 若用KEYBOARD ，則用I，J，K，M控制，ESC跳出。這MENU 可以選擇那一ADVENTURE去CR—EATE，它包括：

RIVERS OF LIGHT
LAND OF AVENTURIA
FANTASY BUILDER SET
SPY & MYSTERY BUIDER SET
SCIENCE FICTION BUILDER SET

若大家是初學者，筆者提意大家創造LAND OF AV—ENTURIA ，它不但教你 HOW TO PLAY ，也是全ACS中最容易和最基本的。它擁有數個經典故事如愛麗斯夢遊仙境。按BUTTON 後放入DATA DISK(因爲ACS擁有數個ADVENTURE，所以筆者不可能盡錄，以下全是以LAND OF AVENTURIA 爲標準)的正面，螢幕上便會打出反白的ACCESSING DISK 表示它正在磁碟中讀出資料，不久，它會叫你放入一面空碟(以上只是在有一部磁碟機的情況，若有兩部磁碟機的用家更方便得多)，經過五次換碟後便會再出現MASTER MENU。

D是COPY 一個ADVENTURE 到另一面DATA DISK，功用和M相似。

C是CONSTRUCT ADVENTURE，它就是ACS之中心部份，它是建立和更改一個ACS或你的ADVENTU—RE ，按"C"後便會出現一個路綫圖，內容如下：

(I) CHANGE NAME

這功能可更改ADVENTURE 的名字。

(II) CHANGE BYLINE

這功能可更改作者名字。

(III) EDIT ADVENTURE INTRODUCTION

這功能可以編輯ADVENTURE初初出現的文字，但要自己打一篇全新的，所以筆者認爲若只玩ACS的ADVEN—TURE，就最好不要改動它，免至用家不明遊戲所玩是甚麼！

(IV) SAVE YOUR WORK AND EXIT

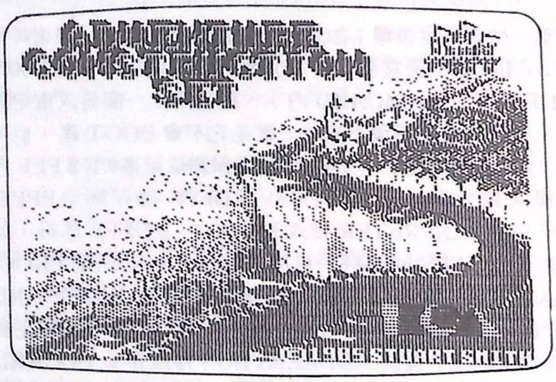
這功能可以令你正在CREATE一面新的ACS碟時，而你又不得閒，可以按此功能而且SAVE入碟及跳出ACS。

(V) SCIENCE FICTION THEME MUSIC

它可以更改ADVENTURE的音樂，如SPY & MYS—TERY及FANTASY 等。

(VI) EDIT WORLD MAP

它可以編輯MAP之分佈、排列等，只要將CURSOR適當的移動便可。



(VII) MOVE THE ENTRANCE TO THE ADVENTURE

這功能可以讓你的主角在MAP中那一處地方出現，這對一個ADVENTURE來說是很重要的，因爲地理位置或地點對於主角去破解是直接影响結果的。

(VIII) DRAW WORLD MAP

這功能是可以畫上或用上ACS本身的圖案，只要用CURSOR將點移動，畫出你創造的圖案。

(IX) CHANGE DEFINITION OF MAP TERRAIN

移動CURSOR使到某地方的圖案更改。

(X) CHANGE PICTURE

這功能可更改MAP上指定圖案爲另一圖案。

(XI) WHICH IS IMPASSABLE TERRAIN

這功能可令指定的圖案變成可出入或指定出入。

創作冒險遊戲大躍進

ACS 創作遊戲大獎

編者註：歡迎讀者利用ＡＣＳ製造個人的ACTION ADVENTURE GAME交來電腦時代，一經採用，即致送日本高級磁碟一盒。所有來稿必須連同磁碟及簡單文字說明。截止日期是一九八五年十二月卅一日。

一切造好後，大家便移動CURSOR 到LET ACS FINISH YOUR ADVENTURE 這一項讓ＡＣＳ分析你所輸入的資料而爲你創造出一個新的冒險遊戲。跟着你會看見另一ＭＥＮＵ，表示你已經到達ACS ADVENTURE WRITER。

ＭＥＮＵ中有幾項前面已說過，ＡＣＳ是讓你作出最後決定的更改，以下說明其中幾個比較特別的部份向大家介紹：

（I）CHALLENGE

這是讓你調校ACS ADVENTURE的LEVEL，1至3是EASY，4至7是中等MEDIUM，8至10是高級HA—RD。

（II）GOAL

這功能是讓你更改所找尋的物件，其中有ＭＥＬＥＥ WEAPON ARMOR MISSICE WEAPON，MAGIC ITEM 和TREASURE 及MAGIC SPELL 等，數目繁多。

（III）ACS COMPLETELY REDRAWS THE MAP

這功能可更改是否由ＡＣＳ重新繪畫另一個ＭＡＰ或者加上數個DOORS到MAP。

（IV）ACS TRIES TO USE ONLY DEFIN—ED THINGS

這功能是讓ＡＣＳ是否只有你所供給的資料或是讓ＡＣＳ加上幾個人物或者是物體。

完成更改後，便到LET ACS WRITE THE AD—VENTURE，跟着你會看見一個反白畫面，跟着ＡＣＳ的轉碟方法去做，不久你便看見ＡＣＳ說要等候二十分鐘，至於ＡＣＳ在CREATE時會以０和＊互相閃動作爲令用者知道電腦正在運行。

但是有一點大家要明白的是，ＡＣＳ所創造的只有AC—TION ADVENTURE，並不是大家比較常見的TEXT ADV.或者GRAPHPIC ADV.。一切完成後，你應該關掉你的電腦電源，因爲創造一個ADV.是十分費時的，你的電腦已十分酷熱，ＡＣＳ都會叫你這樣做。

如果大家玩完了LAND OF ADVENTURIA，大家會發覺除了時間這個問題外，大致上都十分好玩，大家如見有"H"字樣的地方是表示ＨＩＮＴ，還有大家要留意的地方是在遊戲進行中是否對方是你的敵人或是朋友親人等。盡量使本身的HEALTH健康，方法是接近紅十字的圖案時，接觸到它，你的BAR CHART 便會上升，表示你的健康程度有上升。

大家可以嘗試玩玩ＲＩＶＥＲＳ OF LIGHT這個高級的ADVENTURE，背景是在埃及這個地方，老婦人是一個好幫手。

總括來說，ＡＣＳ可算是一個成功的創造ADV.的工具，比起ADVENTURE CREATOR 可以說是無得比。還有一樣是它會檢查一下是否有MOCKING BOARD，比起MUSIC CON. SET要拿起Ｚ－８０咭，技術又進一步。

香港雲生電腦總經銷　日本三菱圖文傳眞機

日本三菱圖文傳眞機之香港總代理——菱電工程有限公司，現委任雲生電腦有限公司爲香港總經銷。雲生電腦有限公司，於一九七九年成立，爲本地微型電腦市場的墾荒者，代理世界名廠電腦產品，及文儀器材，率先引進手提圖文傳眞機。其產品及服務，向受用戶信賴。

日本三菱產品品質優良，由家庭電器、汽車以至重型工業機械等，均得到各界人仕讚賞。

雲生電腦有限公司總裁王榮興先生，對三菱圖文傳眞機極具信心，因爲三菱率先採用新科技，發明工業用的雙色圖文傳眞機。而適合一般辦公室使用的型號包括 MELFAS 5，MELFAS 5150 及 MELFAS FA585G 等。MELFAS 5150 圖文傳眞機更能於十二秒內，將Ａ４呎度文件傳遞至世界各地，爲目前所有座枱式圖文傳眞機中，傳送速度最高之型號。



（圖爲雲生電腦及菱電工程各代表簽約後合照）

王榮興先生深信，憑三菱圖文傳眞機的卓越性能，再加上雲生電腦有限公司的完善、專業銷售後服務，料必能在市場佔一重要席位。

有放影機功能的 FRAME-UP

FRAME-UP 是 BEAGLE BROS 公司較早時（八二年）出品的一個應用程式，不過一直都未有人提及其內在功能，故筆者特借此文對這個程式作一深入的剖析。

FRAME-UP 是什麼

FRAME-UP 這個程式的作用就如放影機（PROJECTOR）一樣，使你能夠順序地展示一些文字或高低解像畫面（FRAME）。至於展示的時候既可以由使用者自行控制，又或是設定好由電腦自動展示每個畫面。

如能好好地運用 FRAME-UP，這個軟件的用途是多方面的。例如把其作爲會議或講座中的輔助工具，又或者利用作橱窗上的訊息顯示亦未嘗不是一個好主意。

特點

FRAME-UP 是以機械語來編寫的，故執行速度非常之快。另外，從磁碟中載入一幅高解像畫面只需2.5秒，較正常的 DOS 足足快了4倍有多。磁碟上的容量亦較多，一面磁碟可以存放17幅高解像畫面或136幅低解像/文字畫面。如有需要，你還能使用兩部磁碟機以展示更多的畫面。

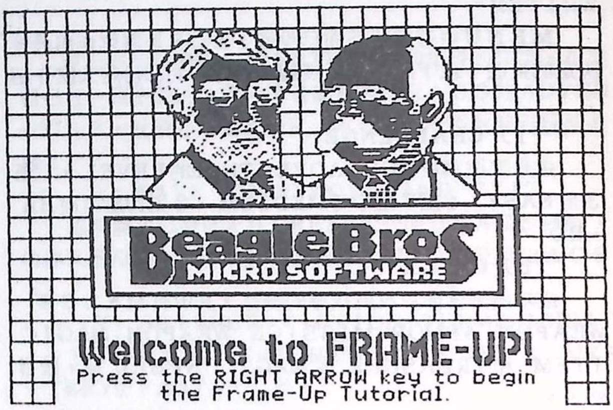
FRAME-UP 的編寫着重於方便及容易使用，包括了單鍵指令及雙向捲動目錄等都旨在方便使用者。至於展示方面，則可以由使用者自行反覆觀看每個畫面，或可以預先設定了每個畫面的展示時間（1至99秒）而作出自動的連續展示，總之就是各適其式。

在製作畫面方面，FRAME-UP 提供了一個 TEXT EDITOR 以作編輯文字檔。其餘的高解像畫面之類，則你需要一個其他的繪圖軟件來協助製作了。

學習使用 FRAME UP

BOOT 起了 FRAME-UP 以後，按〈T〉及〈RETURN〉便可看見一個 TUTORIAL，內容包括了 FRAME-UP 的介紹和使用方法。按左右箭咀或 JOYSTICK 上的 BUTTON（0），（1）可以自由地選擇要看的畫面。由於 TUTORIAL 亦是由一些文字及高解像畫面所組成，故如能由頭至尾看 TUTORIAL 一次定必對 FRAME-UP 所提供的功能加深了解；按〈ESC〉鍵便能離開此狀態。

看過了 TUTORIAL 以後，各位定必想自行製作一組畫面供 FRAME-UP 之用。自然，首先是要製作一隻資料磁碟，在主目錄裡按〈O〉再按〈I〉便可以。但要注意 FRAME-UP 的資料碟存放格式不同於普通的 DOS 3.3，兩者不能互用。附帶一提的是在主目錄後按〈O〉還能選擇磁碟機，改變印字機參數等，最好在作其他工作之前預先設定。



跟着便可按〈I〉或〈A〉以加插一幅畫面，電腦再會詢問加插是一幅什麼的畫面，你可打入：

H—高解像畫面

L—低解像畫面

M—混合畫面

T—文字畫面

選擇了畫面後，須要決定那畫面的來源：

D—從一隻標準 DOS 3.3磁碟取得

F—從一隻 FRAME-UP 資料碟取得

K—直接從鍵盤鍵入（文字畫面）

只要作出了適當的選擇及照着提示去做自能成功地製造一組供 FRAME-UP 使用的畫面了。至於如何把這些畫面排列而使其有系統的展示於別人眼前，則以下會繼續詳述。

FRAME-UP 有兩個展示模式，就是剛才已提及之手動（由使用者控制）和自動（定時展示）模式。在手動模式下，可以用左右箭咀鍵來選擇要看的畫面（使用 JOYSTICK 上的 BUTTON0或1也可以）。此外你還能夠按〈*〉鍵來對畫面編輯，用〈@〉鍵可以傾印一幅畫面，以及〈Λ〉來連結另外的一隻 FRAME-UP 資料磁碟。要離開展示狀態，可按〈ESC〉鍵。

至於自動展示模式，可在目錄中按〈T〉，跟着就會看到一列之時間在右面出現，開始之時這些時間全部都設定爲零。使用左右箭咀來選定一個畫面，便可按入其展示時間（1-99）。除了打入時間外，你還能按入〈Λ〉以使展示到此畫面之時，轉換至另外一部磁碟機，另外如按入的是〈S〉鍵，則使 FRAME-UP 展示到此畫面之時，停止自動展示，而由使用者以左右箭咀及 BUTTON 0，1繼續看其他畫面。把所有要展示的畫面設定好時間後，按〈B〉可開始一個定時的展示了。有一點要補充的是，如在一畫面的展示時間輸入一〈0〉，則該畫面是不會被展示出來。

有放影機功能的FRAME-UP

編輯文字畫面

FRAME-UP 附有一個編輯器來修改文字畫面，在主目錄或手動狀態下按〈*〉便可以看到一個閃動的游標，可按〈CTRL-A〉，〈CTRL-Z〉，左及右箭咀鍵使其上下左右移動，跟着便可對畫面進行修改了。當然，你可以使用其他的文字處理器軟件來製作文字畫面，只要是用 TEXT FILE 來記錄便可以。

傾印畫面

在主目錄或手動狀態下按〈@〉能把顯示中的畫面印出，不過首先要設定印字機所有的參數。要設定參數，可先在主目錄〈O〉再按〈P〉，最後在 TEXT SETUP 及 GRAPH SETUP 根據你的印字機的控制碼打入：

ΛE 代表　ESC

ΛR 代表　RETURN

ΛC 代表　CONTROL 字符，例如 ΛCQ 便代表了 CTRL-Q

ΛH 代表　十六進制數字

注意，不正確的印字機控制碼會導致不能預料的事情發生，所以最好先參考印字機的手册。

注意事項

FRAME-UP 使用 DOS 3.3的高解像畫面要求長度爲 34 SECTORS，而文字或低解像畫面爲6 SECTORS。如不是，可用以下方法轉換：

BLOAD FILENAME，A＄2000

DELETE FILENAME

BSAVE FILENAME，A＄2000，L＄2000

低解像畫面把 L＄2000改作 L＄400。

在定時狀態下可製造出有趣的動畫效果，只要製作一組畫面，而每個畫面只有少許變化，再每幅畫面設定時間爲一便可以了。在 TURTORIAL 亦有一個動畫示範，不過要看此示範先要用左右箭咀指至 ANIMATION，再不斷地按〈T〉及〈B〉便可看到一個動畫效果了。

附帶一提的是在畫面的名字右面是該畫面的種類，如 T 字代表是文字畫面。反相了的 T 是按 S（SKIP）後的結果，反相了的畫面是不會被展示出來的。

最後，結束本文前，筆者要强調，過去專門協助繪畫設訂高解像畫面的軟件不多，所以在製作活動動畫比較困難及複雜，但今時坊間經已有許多這類軟件，希望讀者能利用這些軟件及 FRAME-UP 做齣好戲！■

各位業餘程序編寫員萬勿錯過本文

電腦時代爲你開闢一條新途徑

各位業餘程序編寫員，你們的軟件創作才華有沒有被人忽視？你們的心血作品有沒有人賞識？

我們知道，你們這群業餘程序編寫員絕大部份都是天才橫溢，所創作編寫的軟件並不遜色於任何的職業程序設計師，而且，有許多更是超乎市面要求的商業水準。你們的作品之中，有許多是經年累月的經驗。我們懇請你們不要埋沒自己的天才，將你的經驗創作成果與每一個電腦迷分享。

電腦時代出版社與 COMPUHOUSE 軟件製作所願意爲你全盤策劃出版你的軟件創作成果，讓它流傳到每個電腦用家手中。你絕對不用担心任何的問題，我們會代你効勞一切有關版權、出版、發行製作等問題，你只要安心在家創作便可。

我們擁有豐富經驗的行政人員、軟件工程師、編採出版人材。而且更具有一個廣大的市場發行網，我們的發行網除遍達香港外，還推廣到台灣、星馬泰等東南亞國家，而且更延伸至美國及加拿大。中國大陸這個龐大的市場當然也在我們足跡之下。

透過我們出版軟件，你將會得到最大的收穫。

我們龐大隊伍的生產綫，不單設備完善，且有足夠能力應付任何各方面的需求。

我們歡迎任何機種及類型的軟件創作，無論是ＩＢＭ、ＡＰＰＬＥ電腦、由娛樂軟件、教育應用至商業軟件，只要你們有信心認爲自己的作品有一定的商業水準，我們都樂意爲你出版。

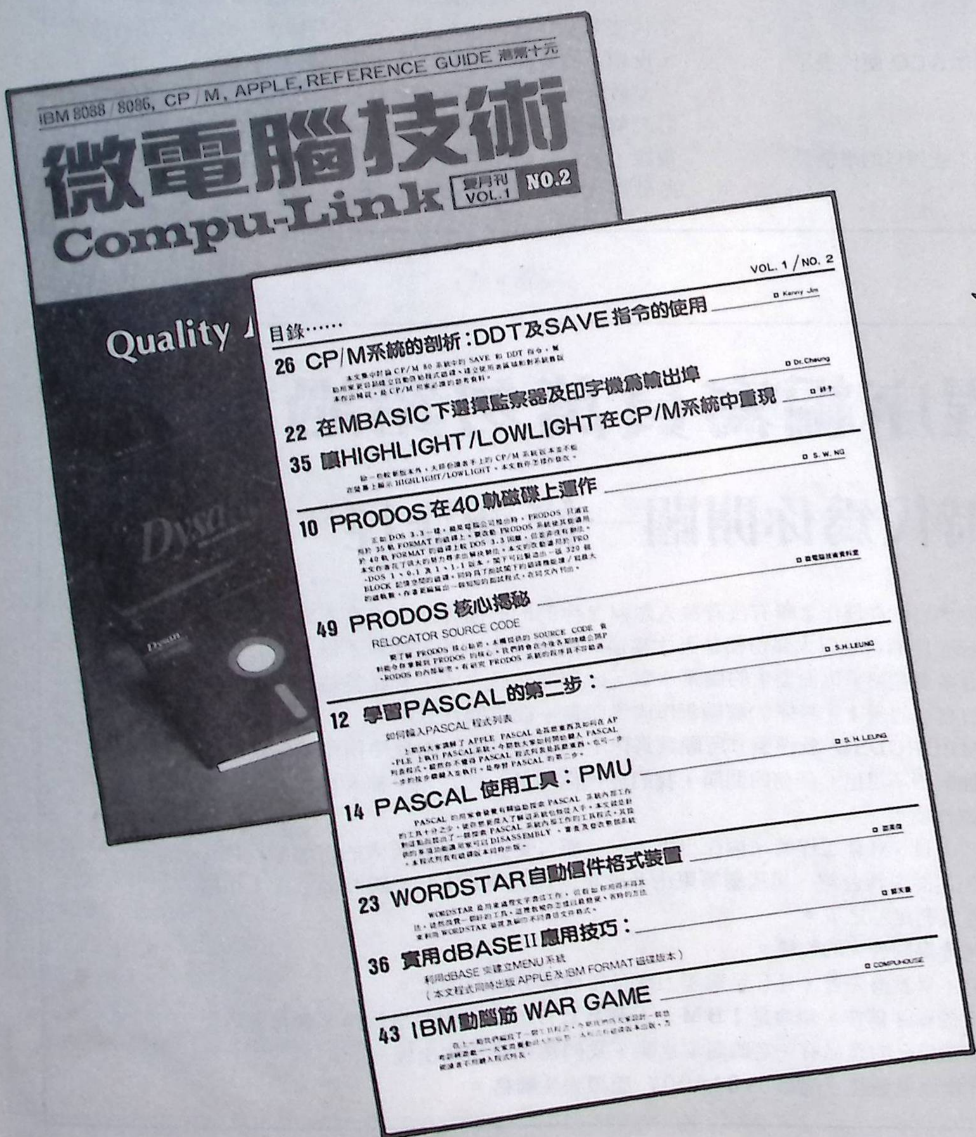
歡迎你們携同作品前來本出版社商談，或電 3—312007 與譚先生聯絡。

ATTENTION PLEASE !!

A NEW REFERENCE GUIDE
FOR SERIOUS COMPUTER USERS / PROGRAMMERS

Now Finally Comes Out !!!

微電腦技術
Compu-Link



第 二 期
九月二十日經已出版
各大書報攤發售

文章篇篇精彩
程式個個實用

程式磁碟版本同時出版

一輯電腦用家

不容錯過之必讀物

GPLE 使用方法深入研究

1. GPLE 的概念
2. 載入及執行 GPLE
3. 編輯模式
4. GLOBAL EDITING (全域性程式編輯模式) LISTING 的控制
5. 大小楷模式輸入
6. 80行咭的相融
7. 鍵盤特置的緩衝設施
8. ESCAPE 功能之設置
9. GPLE 的附隨修正程式
10. 維護GPLE
11. 記憶體配置圖

附操控指令總表

特約經銷處：

- 電腦時代讀者服務部
- 中環三聯書局
- 創業雜誌中心 西環屈地街22號地下（明愛中心對面）
- 深水埗黃金商塲地庫 **21**
- 資訊電腦中心 北角英皇道121號七海商業中心134室

電腦技術應用叢書經已面世

128K RAM咭 應用專輯

第一章　序言
第二章　選購篇
2.1 前言
2.2 買成品之選擇
2.3 自嵌者之選料及裝嵌
2.4 零件表
2.5 規格說明
2.6 總結
第三章　測試及使用篇
3.1 前言
3.2 在DOS 3.3測試
3.3 在CP／M之測試
3.4 在PASCAL 之測試
第四章　硬體改良篇
4.1 前言
4.2 原理概述
4.3 改良步驟
4.4 總結
第五章　INSPECTOR與128K RAM CARD
5.1 系統軟件之使用
5.2 FAST DOS 與 128K RAM CARD
5.3 PRINTER BUFFER
5.4 總結
第六章　硬件篇
6.1 零件概述（4164，3242，74LS 93)
6.2 工作原理

MOCKINGBOARD 技術應用

第一章　MOCKINGBOARD和聲音關係
INIT 開啓 MOCKINGBOARD 程式
LATCH 設定記存器工作副程式
WRITE 輸入資料副程式
RESET 清除記存器數值副程式
TABLE ACCESS 聲音表副程式
第二章　MOCKINGBOARD 基本運用
ENVELOPE PERIOD 和 ENVELOPE SHAPE
第三章　聲音效果的測試
聲音的混合
MUSIC CONSTRUCTION SET
直接影响音樂的變數
鳥鳴，槍聲和背景音樂的混合
海浪聲和背景音樂的混合
第四章　擴大MUSIC CONSTRUCTION SET
一隻磁碟可以存放幾多音樂曲？
反覆依次演奏問題
MUSIC LIST CREATOR
演奏更長的樂曲
第五章　把聲音和音樂加入程式實例
㈠高空擲物DANGER ROAD
㈡空戰鶯魂AIRFIGHT SHOOT OUT
㈢太空侵略者封閉戰CLOSING ALIEN

一九八五年七月底出版

電腦時代出版社

各期程式磁碟磁帶總目錄

1
1 突破傳統的鍵盤
2 捉球遊戲
3 勇闖電腦室
4 迷宮救美
5 將 APPLESOFT 變為 FORTRAN
6 上、下、左、右移動文字畫面
7 測試工具— BOOLEAN 程式
8 騰空更多磁碟空間 一搬移HELLO
9 重現刪除之檔案
10 電腦時代月曆
11 COMPUTEXT 文字處理
12 時日相距計算程式
13 增強英文字彙能力
14 幼童數學四則運算練習
15 過四關

2
1 貨櫃處理
2 沙蟹幻方局
3 巧破密令
4 美術字繪劃器
5 小階字母處理
6 程式轉載器
7 彩色電視／監視器 色彩測試
8 保留遊戲積分記錄
9 數學套件程式
10 磁碟修護工具
11 加送電腦時代月曆 圖像一個

3
1 DOS 的自動同步追踪器
2 動量不滅定律模擬法
3 自製磁碟索引表封套
4 蘋果編曲器
5 幼童認數字初階
6 賽跑遊戲
7 步步追踪
8 DOS KILLER
9 圍棋遊戲
10 巴哈的古典樂章名作
11 旋轉螢幕畫面
12 雜誌目錄 分類歸檔系統
13 神經病醫生一神經鬆弛體操

4
(1) DOS 操作指令簡化器
(2) TEXT FILE 有些甚麼內容？
(3) 立體迷宮
(4) 英文串字練習
(5) 如何以機械語言來修補 DOS
(6) 電腦動畫製作原理。
(7) 豐富設計的學前兒童 教育程式
(8) 應用數學套件程式(二)
(9) DOS WRITER
(10) 穿梭機哥倫比亞號機
(11) 找回失落了的 STEP 及 TRACE
(12) TRACER控制常式
(13) 將 RAM 卡作為磁碟機使用
(14) 黑房計時器
(15) 太陽系模擬系統

5
1 私人約會備忘日曆
2 三度空間顯像系統
3 將 CP／M 檔案轉為 DOS
4 蘋果萬花筒
5 高解像度文字產生器
6 修訂 DOS 操作指令簡化器
7 濃縮圖片儲存空間
8 幼童數學練習── 數字咭
9 加數機大登場
10 備忘 MEMO紙
11 光速計數器
12 剩餘磁碟空 SECTOR數目
13 16進制輸入數字鍵盤
14 問題棋盤
15 神偷智破夾萬
16 隔山打牛
17 大空船降落任務

6
(1) 私人會計系統。
(2) APPLESOFT 擴展版本。
(3) GO]〔 HELLO 的程式
(4) APPLESOFT 偵錯系統。
(5) DOS 指令再修訂版本。
(6) 布倫時鐘。
(7) AMPERSAND 數基轉換器
(8) 完整索引錄。
(9) 砌積木。
(10) 神奇數字方塊。
(11) 亡命飛車。
(12) 迷宮搜秘。
(13) 太空怪侵略戰。
(14) 累加遊戲。
(15) 為磁碟命名。
(16) 任意定義鍵盤。
(17) 捉蟲訓練問題。

7
(1) 會說話的蘋果
(2) 私人財政預算系統
(3) 私人收入會計系統
(4) 高解像字符繪劃器
(5) 圖像轉載器
(6) 螢幕文本編輯處理器
(7) 濃縮圖片程式修訂加強版本
(8) 蘋果杯大賽車
(9) 十六個難度的食鬼遊戲
(10) 高爾夫球遊戲。
(11) 空戰驚魂
(12) 美麗圖案設計家
(13) 更多色彩
(14) 列印的困擾
(15) 十一條空 SECTOR
(16) 捉蟲訓練
(17) 80字行咭例子程式

8
(一) APPLESOFT 滙編器
(二) 最簡單的拆帶程式
(三) 英文打字練習
(四) 編印單個LABEL
(五) 高解像頁處理工具箱
(六) 簡易快速的記憶移動方法
(七) 真正要動腦筋的小學四則運算
(八) 七海霸王 (DEPTH CHARGE)
(九) 原子黑洞
(十) 蛇之噩夢
(十一) 收銀機
(十二) 改良的閃動訊息
(十三) 兩欄索引顯示
(十四) 自動三角函數曲線繪劃系統
(十五) 任意鍵盤（三個）
(十六) 每期捉蟲問題
(十七) 奇妙的 &

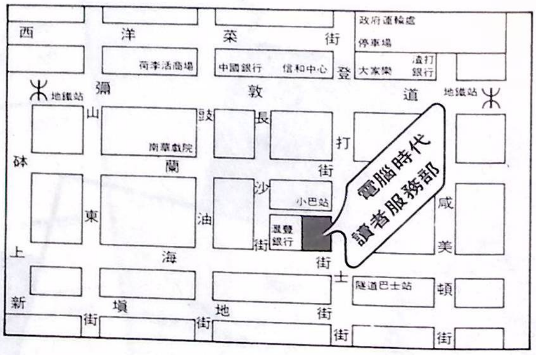
9
(1)磁碟檔案黃頁系統
(2)增強鍵盤輸入及螢幕輸出功能的系統
(3)將工具程式放入DOS佔用的三條磁軌上
(4)快速尋找及磁碟矩陣儲存與提取
(5)新DOS命令改換程式
(6)任意設定目錄索引顯示模式
(7)捉迷藏
(8)古堡之謎
(9)高空擲物
(10)有趣的排序
(11)捉虫問題及答案
(12)第八期捉虫問題答案
(13)印出高解像數目字

10
(1) 可隨意訂定格式之資料紀錄系統
(2) 太空戰士
(3) MARCO鍵盤
(4) 射鴨仔
(5) 奇妙的&──快速分類整理程式
(6) 40軌磁碟工具程式設計
(7) 變數表顯示
(8) 跳越加速器
(9) IQ方塊
(10) 快速檔案搜索
(11) 將DOS移上RAM咭
(12) 培養兒童的記憶能力
(13) 初中代數式運算
(14) 萬用數基轉換器
(15) 十六欄機械碼編印
(16) 捉蟲問題及答案

11
(1) 家庭購物備忘及存貨紀錄系統
(2) 可隨意訂定之資料紀錄系統（下）
(3) 奇妙的＆專欄──矩陣定值
(4) 40軌磁碟工具程式之(二) 單軌 INIT
(5) 兩個自動減省程式佔用之記憶空間應用程式
(6) 對我說
(7) 趣意盈盈的歡樂棒
(8) 英文語句文法分析練習系統
(9) 一行程式示例（共七個）
(10) 美化程式列表
(11) JACKIES HEAD
(12) 魔域迷宮
(13) 巨獸求生記
(14) 自動日子記錄
(15) 空 SEOCTOR 圖表
(16) 捉虫訓練── FACTORIAL！

12
① 家庭開支及個人消費管理系統
② 電子郵箱
③ 40軌程式COPY工具
④ 諾曼第登陸戰（模擬戰爭思考遊戲）
⑤ 工兵掘地雷
⑥ 新改良蛇之噩夢
⑦ 魔域迷宮（APPLESOFT 版本）
⑧ 讓九個月大的嬰孩接觸電腦吧！
⑨ 有多項功能的AMPERSAND工具程式
⑩ 磁頭清潔器
⑪ 程式拯救者
⑫ TAB & XY
⑬ 符號逆輸型（RPN）計數機
⑭ 比RAM DISK更好的RAM咭控制中心
⑮ 圖形建造師
⑯ 捉虫訓練──SINE曲線
⑰ 一行程式

歡迎補購各期程式磁碟或雜誌



精選程式磁碟

編號：CA00A
內容：遊戲及教育程式
遊戲程式
1、捉球遊戲（第一期）
2、勇闖電腦室（第一期）
3、迷宮救美（第一期）
4、過四關（第一期）
5、沙蟹幻方局（第二期）
6、貨櫃處理（第二期）
7、賽跑遊戲（第三期）
8、步步追踪（第三期）
9、圍棋遊戲（第三期）
10、立體迷宮（第四期）
11、穿梭機哥倫比亞號（第四期）
教育程式
12、增強英文匯能力（第一期）
13、幼童數學四則運算（第一期）
14、數學套件程式（第二期）
15、動量不滅定律模擬法（第三期）
16、幼童認數字初階（第三期）
17、豐富設計的學前兒童教育程式（第四期）
18、應用數學套件程式之二（第四期）

編號：CA00B
內容：應用程式精選
1、電腦時代月曆（第一期）
2、COMPUTEXT 文字處理（第一期）
3、美術學繪劃器（第二期）
4、彩色電視／監視器色彩測試（第二期）
5、保留遊戲積分紀錄（第三期）
6、自製磁碟索引封套（第三期）
7、APPLE 編曲器（第三期）
8、雜誌目錄分類歸檔系統（第三期）
9、DOS WRITER（第四期）
10、電腦動畫製作原理（第四期）
11、將 RAM咭作為磁碟機使用（第四期）

編號：CA00 C
內容：私人會計系統
(1) 私人財政預算 （第七期）
(2) 私人收入紀錄 （第七期）
(3) 私人會計程式 （第六期）

編號：CA00D
內容：工具程式集
(1) APPLESOFT程式偵錯系統（等六期）
(2) 將DOS 移上RAM 咭（第十期）
(3) MACRO 鍵盤 （第十期）
(4) RAM 咭控制中心 （第十二期）
(5) 程式編輯器 （第十三期）

每期程式磁碟現仍售29.9元；
精選程式磁碟每張售價45元
(除CA00C 只售30元)。

各期程式磁碟總目錄

13

(1) 行政思考決策輔助系統——成功人士必備工具
(2) 全能標貼製作器
(3) 大力水手——POPEYE
(4) 火樹銀花賀聖誕——平安夜
(5) 低解像圖輔助繪劃器
(6) 達爾文進化論的電腦模擬實踐
(7) 程式編輯器
(8) 磁碟檔案地圖
(9) 增加DOS的CATALOG暫停功能
(10) 新游標
(11) 40軌磁碟改動器
(12) 餅形圖
(13) 自己設信紙版頭
(14) 捉虫問題及答案
(15) 一行程式得獎作品

14A

(1) 高解像圖魔術師
(2) 以軟件來達成的音响合成器
(3) SCREEN SAVER
(4) 磁碟度身定造器
(5) DOS 的快速存取
(6) 一行程式得獎作品選
(7) HIDE：將程式隱藏
(8) 英文字義辨識練習
(9) 數恐龍學習加、減運算
(10) 宇宙城保衛戰（麥法基專欄）
(11) 最後一個立體打井遊戲
(12) 月球一小步，人類一大步
(13) 音量控制
(14) 將APPLESOFT程式轉爲TEXT檔案來編輯
(15) 捉虫訓練——DOS指令的執行

14B

DOS 3.3 FORMAT（SIDE A）
(A) 令蘋果發言的最佳工具MOCKING BOARD SPEECH WRITER
(B) 本港郵費計算諮詢系統
CP／M FORMAT（SIDE B）
(C) CP／M的密碼檔案
(D) 程式設計與FORTRAN所帶來的好處（雙精度計算例子）

歡迎補購各期
程式磁碟磁帶或雜誌

15

(1) 活動電子訊息顯示板
(2) 借貸分期還款計算系統
(3) 麥法基遊戲專欄——DANGER RANGER
(4) 「福」臨香江
(5) 將高解像頁倒轉
(6) 機械碼輸入編輯器
(7) 程序分解擴展器
(8) 利用印字機印出多款字形
(9) 檔案密碼鎖
(10) 兩個新DOS指令之一MREAD
(11) 電腦繪圖專欄——立體繪圖的原理示範程式
(12) 氫氧碳化學分子排列結構模擬
(13) 豬仔錢箱
(14) 3－6－9鬥智遊戲
(15) 超級越軌大賽
(16) 電腦音樂動畫——MONICA
(17) 一行程式入選作品
(18) 捉虫問題及答案

16

(1) 專業室內設計師
(2) 大忙人時間顧問
(3) 學習迷宮產生的原理及自編迷宮遊戲
(4) 使用更方便的BOOT DRIVE 2 軟件
(5) LIST大師
(6) CATALOG HELPER
(7) 40軌磁碟SECTOR使用情況顯示
(8) 磁碟保護器——COPY VTOC法
(9) 兩個新DOS指令MREAD & MWRT（下）
(10) 爲你的程式配上背景音樂
(11) 成雙成對
(12) 吃不到的葡萄不是酸的!!
(13) 動態／策略兼備的戰爭遊戲：BEACH HEAD
(14) 一行程式入選作品
(15) 捉虫問題及答案
(16) 組合線性方程式解答系統
(17) 太空侵略者封閉戰CLOSING－ALIEN

17

(1) 日文書信文字處理系統
(2) APPLESOFT全域編輯器
(3) II PLUS亦可擁有視窗顯示功能
(4) 仿效MOUSE PAINT設計的AP PAINT
(5) 商業發貨單
(6) 把損壞的磁軌鎖住
(7) 配合單軌INIT使用的INIT 40軌
(8) WILDCARD解拆保護原理透視
(9) 兒童串字
(10) 英文聽力測驗
(11) 金字塔數字堆
(12) 麥法基遊戲專欄：TIME PILOT
(13) 武士忠魂
(14) MUSIC CONSTRUCTION SET背景音樂問題
(15) GRAPHIC SOLUTION動畫DEMO
(16) VISICALC篇——DEMO
(17) 一行程式獲選作品
(18) 第十七期捉虫訓練

18

(1) 配合PRINT SHOP使用賀咭信封套印製
(2) 圖像／文字顯示選擇器
(3) 股票投資分析系統
(4) 麥法基遊戲專欄：RAIN SHOWER
(5) 書信文件編打系統
(6) 方尖塔爭奪戰
(7) 步步爲營
(8) VISICALC篇：函數@LOOK UP 的使用技巧
(9) 卡通人物專門店：藍精靈
(10) 磁碟LOCKER
(11) VTOC編輯器
(12) 利用左右箭咀鍵選擇的CATALOG HELLO
(13) 從六合彩到編寫程式技巧
(14) 生動活潑認數目遊戲
(15) 使用BASIC在高解像畫面玩魔術
(16) 機械碼列印對照表
(17) 一行程式入選作品選
(18) 第十八期捉虫問題

19

(1) 電腦時代文件櫃
(2) 辦公室下午茶備忘錄
(3) 蘋果超級500圈大賽車
(4) 五子連環（棋類遊戲百種之三）
(5) 高空救人
(6) 萬用輸入裝置
(7) 卡通人物專門店：小浣熊
(8) 配合卡通人物編印的月曆程式
(9) 時間教授
(10) 十進制換算表
(11) 利用MOCKING BOARD作時鐘咭
(12) 認識一張適用於蘋果機的中文咭
(13) 自動檢定磁碟質素的DISK AID
(14) 三種美麗的展示高解像圖方法
(15) 直接讀取GAME FILE：活用RWTS
(16) 一行程式入選作品
(17) 第十九期捉虫問題

20

(1) 快速DATA處理程式
(2) 印字機功能大揭秘
(3) 中文咭應用專欄——生肖年份檢索
(4) 電子線路設計工具套件
(5) 另一個功能鍵盤程式——POWER KEY
(6) 麥法基遊戲專欄：DEATH RACE
(7) 棋類遊戲百種之四——四位一體
(8) 暗渦截擊戰
(9) 麥法基專欄：創造一個移動的立體背景DEMO
(10) 機械語言專欄：程式執行速度的探討
(11) HELLO程式漫談／HELLO BREAKER
(12) 觀看文字檔案
(13) 輔助程式編輯器
(14) 從足球運動中學習物理現象
(15) 看圖學英文生字
(16) 從文字幕到高解像頁
(17) 改良TRACK & FIELD

21

(1) 模擬印字機BUFFER咭
(2) 萬用統計圖表製作器
(3) 核子螞蟻
(4) 棋類遊戲之五——海戰棋
(5) 麥法基創作遊戲——STAR WAR
(6) PRODOS 專欄——PROCAT
(7) 機械語言應用——SLIDE SHOW系統
(8) 有七項功能的DOS 設計器
(9) 程式執行監察器
(10) 文字幕旋轉器
(11) 函數擴展及因子分解
(12) 智力棋——TESTMIND
(13) 逐漸擴大方孔式畫面顯示

電腦時代合訂本新增內容程式磁碟

電腦時代精裝合訂本（1－12期）
新增內容程式磁碟集：
(1) 將低解像圖或文字頁存入RAM咭
(2) 將高解像圖頁存入RAM咭
(3) 磁碟操作系統專論－RWTS 原程式
每張只售19.90
歡迎到讀者服務部購買或利用郵購方式。郵購辦法請參閱郵購需知欄。

電腦時代
分類廣告

「電腦時代」由第由21期開始，增設分類小廣告欄歡迎各界賜稿惠刊。

每段小廣告：**大字10個，小字50個，收費30元。**

＊來稿請繕寫清楚，免有校誤。
＊凡涉及法律性廣告，請携同身份證。
＊每月十號前截稿；來稿可直接交來「電腦時代」讀者服務部，或以劃線支票（抬頭填寫「電腦時代出版社」）郵寄九龍中央郵政信箱71193號。
＊廣告費請先惠付（超過三期九折優待；超過六期者，更獲八折優待），刊字如超過一段照兩段計。
＊每一廣告客刊，最多只能刊登三段。

____________先生/小姐/女士/公司

茲附上港幣__________元（現金/支票№__________）請將附上之廣告代爲刊登；由「電腦時代」_____期開始，至_____期止，合共_____則。此致

電腦時代出版社

刊戶：____________________

地址：____________________

電話：__________ 日期：__________

電腦時代程式磁碟

每張只售HK$29.90

電腦時代每期都刊登多個精彩的程式，其中必然有你喜歡的。當你讀完文章興緻勃勃的坐下來欲進一步深入了解程式的精彩處時，看見那大堆冗長的列表，就甚麼興趣也沒有了！那麼，你爲何不購買一張「電腦時代程式磁碟」呢！

每期出版的電腦時代所登載的程式，均全部紀錄入一張高品質的磁碟上，並隨雜誌同時出版。爲省卻打入程式的麻煩，徒然浪費你寶貴的時間，免除了花精神捉錯打字的痛苦，今日就請就近的特約經銷處或電腦時代讀者服務部/COMPUHOUSE購備一張。

程式磁碟每張只售港幣 29.90 元。

第22期程式磁碟內容

*All Programs are in Apple version

(1) PRINTSHOP 助手——印製私人名片及信紙
(2) 活用印字機
(3) 把機械語化成BASIC
(4) 從程式中學習編寫中文咭程式技巧——砌唐詩遊戲
(5) 唱片櫃管理系統
(6) 電子收銀／計數機
(7) 16K WILDCARD
(8) 智力棋的考驗——MASTER MIND
(9) 德國橋牌
(10) 麥法基創作遊戲——立體死亡大賽車
(11) PRODOS 專欄——DATA・SET
(12) 磁碟壽命性能測試器
(13) 機械碼輸入錯誤核對工具
(14) 排序原理透視
(15) CRAZY PROTECTION
(16) LOAD RUNNER MAKER
(17) 80字行兩欄目錄檔案顯示

本期程式磁碟雙面灌錄　讀者留意

爲方便讀者購買程式磁碟，特新增香港電視服務站爲我們特約經銷處；現前往各大地鐵及九廣鐵路電視服務站均有代售。

香港電視服務站

太古城服務站

太古城中心（太古城道）

地鐵站

荃灣服務站	樂富服務站
葵芳服務站	九龍塘服務站
美孚服務站	石硤尾服務站
荔枝角服務站	旺角服務站
長沙灣服務站	油麻地服務站
觀塘服務站	佐敦服務站
九龍灣服務站	尖沙咀服務站
彩虹服務站	金鐘服務站
黃大仙服務站	中環服務站

九廣鐵路車站

九龍塘服務站	大圍服務站
旺角服務站	沙田服務站

中環

百老滙攝影器材有限公司（第五店）
域多利皇后街 3 號（中環街市側）
電話：5－235513

亞富影音
士丹利街 28 A（陸羽茶樓側）
電話：5－240723

三聯書店讀者服務中心
香港中環域多利皇后街九號（中環街市側）
電話：5~250102

銅鑼灣

百老滙攝影器材有限公司（第十四店）
軒尼斯道 529號地下（三越百貨公司對面）
電話：5－770181

太古城

百老滙攝影器材有限公司（第七店）
太古城中心第 232店
電話：5－672006

西環

創業雜誌中心
西環屈地街22號地下　明愛中心對面
電話：5-478338

旺角

百老滙攝影器材有限公司　（第 1,3,9,10 店）
彌敦道 758號在下（凱聲戲院側）
電話：3－816931
彌敦道700 號旺角中心第二期地下
電話：3－943827
豉油街 26 號地下（近商務書局）
電話：3－884331
亞皆老街 60 B 地下（世界書局對面）
電話：3－956367

深水埗

百老滙攝影器材有限公司（第十三店）
青山道148 號地下（華聲戲院對面）
電話：3－7295938

尖沙咀

智源書局
金巴利道 27—33號永利大厦 2字樓A座
電話：3 － 678482 3－678414

荔枝角

百老滙攝影器材有限公司（第二店）
美孚新邨第四期百老滙街 79 號（影都戲院側）
電話：3－7424923

觀塘

百老滙攝影器材有限公司（第四、十二店）
物華街 3 號地下（皇上皇側）
電話：3－419513
同仁街 35 號地下（大家樂側）
電話：3－449940

荃灣

百老滙攝影器材有限公司（第十一店）
沙咀道 230 A（戴麟趾夫人健康院對面）
電話： 0-4938926

北角

資訊電腦中心
北角英皇道121 號七海商業中心 134 室
電話：5-708301

高登/黃金商場特約經銷處：
深水埗福華街黃金商場地庫21號

教育專業人員協會（北角會所）

電腦時代讀者服務部／CompuHouse

地址：旺角上海街438號同珍商業中心11字樓1104室
（上海街／登打士街交界）　電話：3-312007

補購各期程式磁碟可前往讀者服務部／COMPU HOUSE

Apple is the trademark of Apple Computer Inc.

電腦時代精裝珍藏本（1-12期合訂本）

隨書附送價值超過 150元的購買程式磁碟優待券

精裝珍藏本，顧名思義，是將過去出版的電腦時代第1至12期所有的精彩內容滙集成一巨册，其中包括超過一百個的軟件程式使用說明，數十篇有關各類週邊硬件深入解剖報告、熱門軟件評介及用法文章。是所有擁有 APPLE或 APPLE兼容電腦的用家必須珍藏的巨獻。

可能你已擁有第一至十二期，但這册精裝珍藏本依然有你值得擁有的價值，因爲我們在這册合訂本中加插了多篇從未刊登過的文章及程式。而且我們更隨書附送價值超過150元的購買程式磁碟優待券。

電腦時代精裝珍藏本全書厚度超過200餘頁，採用穩固的膠背釘裝。每册僅售50元。

新內容

將低解像圖或文字頁存入RAM咭 16
將高解像圖頁存入RAM咭 18
將APPLE的編輯功能提升到大機水平的軟件GPLE 29
PRODOS兩頁GLOBAL PAGE深入探討 27
磁碟操作系統專論(一)——穿梭磁碟中 20
磁碟操作系統專論(二)——RWTS深入探討 25

週邊硬件探討篇

如何選購AP II電腦 31
AP II微型電腦之進化史——六代同堂 34
AP II微電腦外展插座之功能及應用 37
門外漢選購印字機 40
80字行咭縱橫談 43
再談80字行咭 44
Z 80咭的認識 47
再談Z 80咭地址配對方法 50
16 K RAM 咭潛能發揮的戰鬥技巧 53
只花三元改裝16 K RAM咭增加狀態顯示燈 55
加速APPLE運作速度的多種方法探討 57
妙用無窮的128K RAM咭 60

軟件縱橫談

如何用FID抄錄被保護之磁碟 71
漫談6502十三種存取資料地址模式 74
RESET鍵的特殊問題 78
將WORDSTAR放入RAMDISK使用(128 K RAM咭之一) 83
令你的APPLE DOS更多彩多姿 81
改良WORDSTAR加強編印字體款式效果 84
有廿四種功能的DOUBLE TAKE 86
APPLE TOOL KIT各項優異功能 89
初學組合語言者的福音——LISA 2.5細談 92
速成學會VISICAL（上） 96
速成學會VISICAL（下） 99
PRINT SHOP印圖功能的大躍進 102
電子辭典的典範——SENSIBLE SPELLER 105
程式保護之父——LOCK IT UP的使用方法 107
NIBBLE AWAY（BI）的小小秘密 106
專業級的文字處理器WORDSTAR 110
卓越的繪圖工具E-Z DRAW 111
聲音和畫面的技巧——用GRAPHICS MAGICIAN編寫的遊戲 68

DOS及PRODOS探討欄

磁碟的結構認識 113
■ 磁碟修護工具 120
DOS BOOT碟的原理 122
■ 如何以機械語言來修補DOS 125
■ 用機械語言直接控制DOS指令 127
■ 將工具程式安放入DOS佔用的三條磁軌上 128
■ 單軌INIT 40軌磁碟工具程式之一 91
PRDOS縱橫——使用方法介紹 131
PRDOS縱橫——磁碟格式 134
PRDOS縱橫——啟動過程及記憶體之使用技術 133
在APPLE相容的電腦上執行PRDOS 135
蘋果最新操作系統PRDOS介紹(上) 136
蘋果最新操作系統PRDOS指令介紹(下) 139

應用軟件庫

■ 電腦時代月曆 142
■ COMPUTEXT文字處理 143
■ 彩色電視/監視器分色彩測試 145
■ 製磁碟索引表封套 142
■ 雜誌目錄分類歸檔系統 149
■ 神經病醫生——神經戰強體操 151
■ 將RAM卡作為磁碟機使用 152
■ 廚房計時器 135
■ 私人約會備忘日曆 157
■ 將CP/M檔案轉為DOS 200
■ APPLESOFT擴展版本 158
■ 私人會計系統 163
■ APPLESOFT程式語法偵錯工具 165
■ 會說話的蘋果 166
■ 私人財政預算系統 168
■ 私人財政收入系統 170
■ APPLESOFT運編器 172
■ 磁碟檔案資料系統 174
■ 增強鍵盤輸入及螢幕輸出功能的系統——COMPUKEYSOFT 176
■ 可隨意訂定格式之資料紀錄系統 178
■ 可隨意訂定格式之資料紀錄系統(下) 181
■ MACRO鍵盤 182
■ 語音加速器 184
■ 家庭購物備忘及食物用品存貨紀錄系統 186
■ 家庭開支及個人負債管理系統 188
■ 電子郵箱 189

工具程式篇

■ 美術字繪畫器 190
■ 旋轉螢幕畫面 180
■ DOS殺手 167
■ DOS的自動同步追蹤器 192
■ DOS操作指令簡化器 95
■ 自動加插DOS指令入程式中——DOS WRITER 126
■ TEXT FILE有些什麼內容？ 121
■ 濃縮圖片儲存空間 185
■ GOTO一個取代HELLO的程式 194
■ DOS指令再修訂版本 195
■ 圖片轉載器 124
■ 螢幕文本編輯處理器 175
■ 高解像字符繪畫器 237
■ 高解像頁處理工具篇 171
■ 編印單張LABEL 150
■ 任意設定目錄索引顯示模式 189
■ 將DOS移上RAM咭 195
■ 十六進制碼編印 197
■ 自動減省程式長度 193
■ 美化程式列表編印 141
■ RAM咭控制中心 196
■ 簡化繪製圖形表過程——圖形建造師 198
■ LOCK IT UP的破解方法 198
■ 有多項功能的AMPERSAND工具程式 199
■ 蘋果萬花筒 242

中學及兒童教育軟件集

■ 增強英文字運用能力動動腦筋串生字 229
■ 幼童數學四則運算練習 237
■ 數學套件程式 226
■ 動量不滅定律 235
■ 幼童認數字初階 229
■ 英文串字練習——加強兒童識字能力 237
■ 豐富設計的學前兒童教育程式 230
■ 應用數學套件程式(二)——三項運算功能 231
■ 幼童數學練習——數字咭 232
■ 加數練大會場 233
■ 神奇數學方塊 148
■ 砌積木 233
■ 自動三角函數曲線繪畫系統 234
■ 真正會動腦筋的小學四則運算 236
■ 培養兒童的記憶能力 233
■ 初中代數式運算 237
■ 常用數基轉換器 224
■ 告訴我 230
■ 神奇蛋蛋的對換格 238
■ 英文語句文法分析練習系統 187
■ 符號逆轉型（RPN）計數機 239
■ 讓九個月大的嬰孩操縱電腦吧！ 219
■ 太陽系模擬系統——探測六大行星移動位置 240

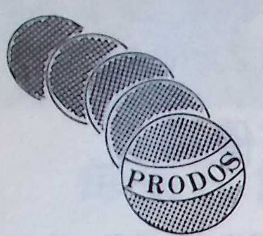
娛樂遊戲軟件集

■ [illegible] 204
■ 桌球遊戲 197
■ 樂聲電腦室 104
■ 迷宮救美 82
■ 音樂處理 201
■ 小蟹幻方陣 202
■ 賽跑遊戲 109
■ 步步為營 214
■ 圍棋遊戲 191
■ 立體迷宮 204
■ 神偷智破大厦 205
■ 太空船降落任務 220
■ 迷宮探險 138
■ 太空爭霸戰 73
■ 亡命飛車 173
■ 累加遊戲 201
■ 蘋果杯大賽車 206
■ 十六個難度的食鬼遊戲 218
■ 高爾夫球遊戲 208
■ DEPTH CHARGE（七海霸王）（上） 209
■ DEPTH CHARGE（七海霸王）（下） 215
■ 孫子兵法 49
■ 古堡之謎 217
■ 填空遊戲 101
■ 探迷宮 219
■ 太空戰士 220
■ 射鴨仔 85
■ IQ方塊 110
■ JACKIES HEAD 221
■ 龍城迷宮 224
■ 打蟹先生 225
■ 數砌板之遊戲 59
■ 工兵掘地雷 208

特約經銷處：

- 電腦時代讀者服務部
- 中環三聯書局
- 教育專業人員協會（北角會址）
- 深水埗黃金商場地庫21

第一套UTILITY: PROBYTER



PRODOS 系統面世經有年半，但由於有關的參考資料不多，所以PRODOS 未能像DOS 3·3 般被人深入了解。至於PRODOS 的 UTILITY 程式更是少見。BEAGLE BROTHER 的PROBYTER 是第一個。

PRO－BYTER 並不是一個程式，是一組的工具應用程式，包括有：

(1)BYTEZAP ·PRO
(2)TYPE
(3)CATA · LOG
(4)DATE · SET
(5)LIST · LOCK
(6)CALL · BASIC
(7)SCRAMBLED · BASIC
(8) QSORT
(9)MEM · ZAP

除了這些工具程式外，還有BEAGLE BROTHERS 慣有的APPLE TIPS& TRICKS 及一些PRODOS 的MESS－AGE · DECODER。

BYTEZAP · PRO 主要是一個DISK－INSPECTOR 程式，由多個子程式組成。用途主要是檢視 PRODOS FO－RMAT 的磁碟。在這裏，筆者不打算講述此UTILITY的使用方法，大家只要參考PROBYTER 的使用手冊。（或者日後有機會再另闢一章爲大家詳述。）

TYPE 程式爲PRODOS 加添一個新指令，主要用來將TEXT FILE DUMP 出螢幕或印字機。留意PRO－BYT－ER 上所附隨的SYSTEM是 1 · 1 · 1 版本。此TYPE 程式亦只適用於 1 · 1 · 1 版本。

CATA · LOG 程式主要用來提供CATALOG 時的一些方便。

DATA · SET 程式是可讓沒有CLOCK CARD 的用家打入CURRENT TIME 及DATE 。本文末將會把這程式介紹出來。

LIST · LOCK 程式可將FILE 鎖上，不被其他人LIST 。不過這只是暫時性，BOOT 機後便消失作用。

CALL · BASIC 主要是用來將一行的APPLESOFT 程式轉爲一個可用CALL 指述執行的子程序。此子程序在合理範圍內可任意存放在記憶區裏。

SCRAMBLED · BASIC 是一個示範APPLESOFT 程式指令執行的程式。本程式只適用於 VERSION 1·1·1 。

QSORT 是一個用機械碼寫成的快速SORT 程式，能在 1 秒內將一百個字母排序。

MEM · ZAP 是用來INSPECT MEMORY 內容以便進行EDIT 。

總括來說，這九個工具程式足夠閣下對PRODOS 系統初步探索之用。

在結束本文前，爲大家介紹 PRO－BYTER 內的一個非常實用的短程式：DATE · SET 。

大家都會知道，在PRODOS 系統下製造一個FILE ，假如機內置有CLOCK咭，則會在FILE 後自動加上日子時間記錄。不過，並非每個用家都有CLOCK咭。大家只要執行這個DATE · SET 程式，電腦便會要求你輸入即時的時間日子，然後在FILE 旁出現代替了"（NO DATE）" 。除非大家重新RUN DATE · SET 這程式，否則電腦記憶裏一直會保留剛輸入的日子時間。

DATE · SET 可作爲一個STARTUP程式，如此一來，每回BOOT 機，電腦便會詢問當時的時間。（只要將DATE · SET 改爲START UP 即可）

大家鍵入本程式後，可用SAVE DATE · SET 指令存入磁碟。或者大家可在即期的程式磁碟中找到同一名字的FILE ，但注意，未執行程式磁碟上的FILE 之前，必須要將本檔案轉爲PRODOS FORMAT，否則會出現ERROR 。■

```
1  REM  ***************
2  REM  *  DATE.SET  *
3  REM  *     BY     *
4  REM  * BEAGLE BROS*
5  REM  ***************
10  TEXT : PRINT  CHR$ (21): HOME
   : NORMAL : VTAB 8
20  PRINT "CURRENT DATE AND TIME:
   "
50 MX$ = "JANFEBMARAPRMAYJUNJULAU
   GSEPOCTNOVDEC?????????"
60 A =  PEEK (49040):B =  PEEK (4
   9041):MIN =  PEEK (49042):HR
    =  PEEK (49043)
70 YR =  INT (B / 2):MO =  INT (A
    / 32) + 8 * (B - YR * 2):DA
    = A -  INT (A / 32) * 32
75  IF YR > 99 OR MO > 12 OR MO <
    1 OR DA > 31 OR DA < 1 THEN
    PRINT "<NO DATE>": GOTO 90
80  PRINT  CHR$ (48 * (DA < 10));
   DA"-" MID$ (MX$,MO * 3 - 2,3
   )"-" CHR$ (48 * (YR < 10))YR
   " " CHR$ (48 * (HR < 10))HR"
   :" CHR$ (48 * (MIN < 10))MIN

90  ON EN GOTO 510: PRINT : PRINT
   "NEW DATE AND TIME:"
100  VTAB 11: PRINT : CALL  - 958
   : PRINT "DD-MMM-YY HH:MM": PRINT
   : PRINT "<RETURN> WHEN FINIS
   HED": PRINT "<LEFT ARROW> TO
    START OVER": PRINT "<ESC> T
   O QUIT": VTAB 12: HTAB 1:HT =
   1:X$ = ""
110  POKE 1403,HT - 1: HTAB HT: GET
   A$: ON A$ =  CHR$ (8) GOTO 1
   00: ON A$ =  CHR$ (13) GOTO
   200: IF A$ =  CHR$ (27) THEN
   510
```

```
120  ON (HT < 3 OR HT > 7) AND (A
     $ < "0" OR A$ > "9") GOTO 11
     0: ON HT = 1 AND A$ > "3" GOTO
     110: ON (HT = 11 OR HT = 14)
      AND A$ > "5" GOTO 110: ON (
     HT > 3 AND HT < 7) AND A$ <
     "A" OR A$ > "Z" GOTO 110: ON
      LEN (X$) > 10 GOTO 110
130  IF HT = 4 THEN MO$ = A$: FOR
     MO = 1 TO 12: IF MO$ <  > MID$
     (MX$,3 * MO - 2,1) THEN  NEXT
     : GOTO 110
132  IF HT = 4 THEN  IF MO$ = "F"
      OR MO$ = "S" OR MO$ = "O" OR
     MO$ = "N" OR MO$ = "D" THEN
     X$ =  LEFT$ (X$,2) +  MID$ (
     MX$,3 * MO - 2,3): PRINT  MID$
     (MX$,3 * MO - 2,3);:HT = HT +
     3: GOTO 170
140  IF HT = 5 THEN MO$ =  RIGHT$
     (X$,1) + A$: FOR MO = 1 TO 1
     2: IF MO$ <  >  MID$ (MX$,3 *
      MO - 2,2) THEN  NEXT : GOTO
     110
142  IF HT = 5 THEN  IF MO$ = "AP
     " OR MO$ = "AU" OR MO$ = "JA
     " THEN X$ =  LEFT$ (X$,2) +
      MID$ (MX$,3 * MO - 2,3): PRINT
      CHR$ (8); MID$ (MX$,3 * MO -
     2,3);:HT = HT + 2: GOTO 170
150  IF HT = 6 THEN MO$ =  RIGHT$
     (X$,2) + A$: FOR MO = 1 TO 1
     2: IF MO$ <  >  MID$ (MX$,3 *
      MO - 2,3) THEN  NEXT : GOTO
      110
160  PRINT A$;:X$ = X$ + A$
170  ON  LEFT$ (X$,2) > "31" OR (
      LEN (X$) = 2 AND X$ < "01")
      GOTO 100:HT = HT + 1 + (HT =
     2) + (HT = 6) + (HT = 9) + (
     HT = 12)
180  IF HT = 8 THEN N = 29 + (MO <
      > 2):N$ =  STR$ (N): IF (MO
      = 2 OR MO = 4 OR MO = 6 OR
     MO = 9 OR MO = 11) AND  LEFT$
     (X$,2) > N$ THEN X$ = N$ +  RIGHT$
     (X$,3): HTAB 1: PRINT N$;
190  GOTO 110
200 DA =  VAL ( LEFT$ (X$,2)):MO$
     =  MID$ (X$,3,3):YR =  VAL
     ( MID$ (X$,6,2)):HR =  VAL (
      MID$ (X$,8,2)):MI =  VAL ( MID$
     (X$,10,2))
250  IF  LEN (X$) <  > 11 THEN 10
     0
400  IF YR > 99 OR MO > 12 OR MO <
     1 OR DA > 31 OR DA < 1 THEN
      PRINT  CHR$ (7);: GOTO 100
500 POKE 49041,YR * 2 + (MO > 7)
     : POKE 49040,32 * (MO -  INT
     (MO / 8) * 8) + DA: POKE 490
     43,HR: POKE 49042,MI:EN = 1:
      VTAB 8: HTAB 1: GOTO 20
510 VTAB 13 - 4 * EN: PRINT : CALL
     - 958: PRINT : PRINT "TYPE
     " CHR$ (34)"RUN" CHR$ (34)"
    TO MAKE A CHANGE."
```

在兼融機上 BOOT PROBYTER

曾艮勇

編者註：有關 PROBYTER 軟件的詳介，可參閱同期另文［PROBYTER］

BEAGLE BROS 公司出品之軟件，無須我介紹大家都知非常好用，種類又多，最主要爲所有出品都無什麼保護，這點最合本地那麼多之 APPLE COMPATIBLE 機了。

本年五月 BEAGLE BROS 公司又有新出品了，爲了配合 APPLE 之最 POWERFUL 的 DOS，PRO-DOS。它這個新軟件也是用 PRO-DOS 的，此爲 BEAGLE BROS 公司爲 APPLE PRO-DOS 編寫的第一個軟件，同時亦爲 PRO-DOS 增加了很多它所無之功能。這個新軟件名爲“PRO-BYTER”此軟件主要之功能爲“加强或增加 PRO-DOS 所沒有的功能”。但這個 PRO-BYTER 和 APPLE 公司之 PRO-DOS 一樣，同時要求 ROM F8，即地址 $FB09處之字串爲 APPLE Ⅱ，否則和非 APPLE 之相融機 BOOT PRO-DOS 一樣，BOOT 得一半就整架機停了下來，不再將 BASIC，SYSTEM 和 STARTUP 檔載入。解決的辦法和“電腦時代第十一期”四十二頁，介紹的方法差不多，只是地址不是而已。

在 APPLE COMPATIBLE 機上執行 PRO-BYTER 的改法。

①首先放入 PRO-BYTER 磁片 BOOT 機，當螢幕出現

APPLEⅡ

PRO-DOS 1.1.1　　18-SEP-84

等 TITLE 時，電腦會像 HANG 機一樣，一切停了下來時，按 CTRL-RESET 還入 MONITOR。

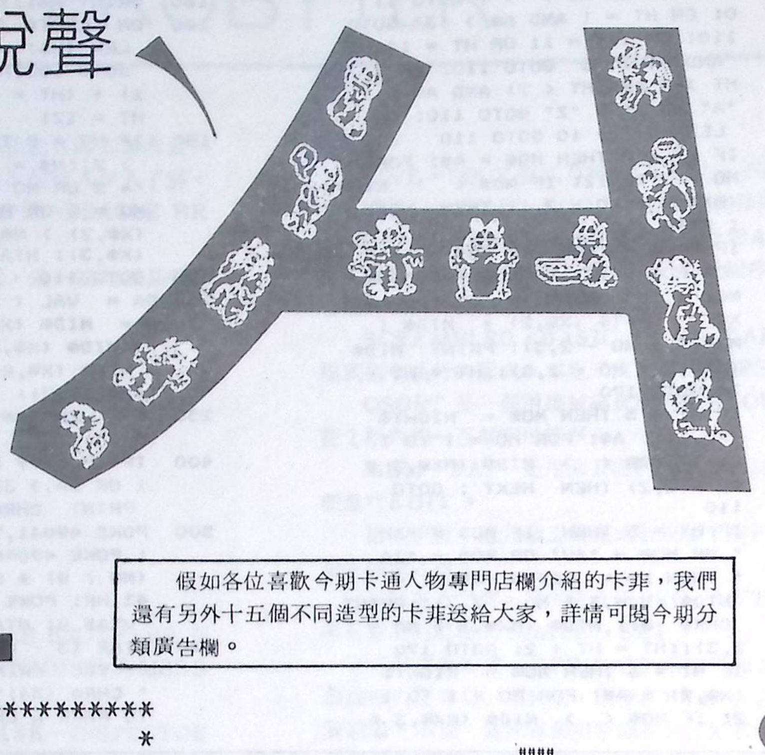
②按 $267CL，把 $269E-D0 03即 BNE $26A3改爲 $269E：EA EA 即 NOP NOP。（即是不要 $269E 的 BNE $26A3）。

③再 $2000G，就可正常地 BOOT 入和執行 PRO-BYTER

如不想每次 BOOT 機時都要改 $269E 處之值，如此麻煩。那可用任何有 SECTOR EDIT 之 COPY PROGRAM 把 PRO-BYTER 磁片之 TRACK 1，SECTOR 9，BYTE 9E 和9F 之值 D0 03改爲 EA EA。以後 BOOT 機就免麻煩了。■

卡通人物專門店

加菲和你說聲



□黃博思□

今期爲大家介紹的公仔是許多人的心愛偶像，千萬巨星——加菲。各位可依照本文列表打入電腦即可。

筆者也爲程式加入一項自動檢錯功能，它並非用CHECKSUM 的，而是當遇到錯誤，便印出當時讀到的DATA行號，和讀入的兩個資料，X和Y。

Y是字符碼，各位請參照程序的行75—100的ASC字符表，而X則是以Y字符印出多少個字。

相信以上的設計是極之方便的。■

假如各位喜歡今期卡通人物專門店欄介紹的卡菲，我們還有另外十五個不同造型的卡菲送給大家，詳情可閱今期分類廣告欄。

```
10   REM  ************************
15   REM  *                      *
20   REM  *    GRAFIELD PRINT    *
25   REM  *                      *
30   REM  *  BY : WONG POK SZE   *
35   REM  *                      *
40   REM  *  COPYRIGHT C- 1985   *
45   REM  *                      *
50   REM  *    COMPUTING AGE     *
55   REM  *                      *
60   REM  ************************

65   CLEAR : TEXT : HOME : DIM A(5
     )
75  A(00)  =   ASC (" ")
80  A(01)  =   ASC ("#")
85  A(02)  =   ASC ("!")
90  A(03)  =   ASC (".")
95  A(04)  =   ASC (":")
100  A(05)  =   ASC ("O")
105   PR# 1: PRINT  CHR$ (24): PRINT
      CHR$ (27)"1"
115   PR# 1: ONERR  GOTO 185
120   PRINT
125   READ X,Y
130   IF X = 0 THEN 155
135   FOR Z = 1 TO X
140   PRINT  CHR$ (A(Y));
145   NEXT Z:D = D + X
150   GOTO 125
```



WRITTEN BY : WONG POK SZE

加菲和你說聲 Hi !

```
155  IF Y > 0 THEN 170
160  PRINT :D = 0
165  GOTO 125
170  IF Y < D THEN  PRINT :D = 0
175  PRINT  SPC( Y - D);:D = Y:N =
     N + 1
180  GOTO 125
185  FOR I = 1 TO 5: PRINT : NEXT
   :ER =  PEEK (222): IF ER <  >
     42 THEN 235
190  PRINT  TAB( 4); CHR$ (27);"4
     ";"WRITTEN BY : WONG POK SZE
     ": PRINT  CHR$ (27);"5"
195  FOR I = 1 TO 7: PRINT : NEXT
     : PR# 0
200  PRINT "Do you want another o
     ne(Y/N) ?"
205  PRINT : PRINT : PRINT
210  GET A$
215  IF A$ = "Y" THEN  RUN
220  IF A$ = "N" THEN 230
225  GOTO 210
230  PRINT "Good Bye !!!": PRINT
     : VTAB 22: END
235  PR# 0: PRINT : PRINT "ERROR
     CODE ";ER;" AT LINE "; PEEK
     (218) +  PEEK (219) * 256: PRINT
     : PRINT "DATA LINE =";(300 +
     N);"  DATA X=";X;"  DATA Y="
     ;Y: PRINT : PRINT : END
240  REM  ****************
245  REM  * PROGRAM DATA *
250  REM  ****************
301  DATA 0,36,4,1
302  DATA 0,25,5,1,5,0,7,1
303  DATA 0,24,8,1,2,0,7,1,1,2,1,
     1
304  DATA 0,23,7,1,2,2,1,1,1,0,8,
     1,1,2,1,1
305  DATA 0,23,1,1,1,2,1,1,1,2,1,
     1,1,2,2,1,2,2,2,1,1,2,1,1,1,
     2,1,1,1,2,3,1,1,2,1,1
306  DATA 0,23,1,1,1,2,1,1,1,2,1,
     1,2,2,2,1,2,2,1,1,1,2,1,1,1,
     2,1,1,2,2,2,1,2,2,2,1
307  DATA 0,23,1,1,1,2,1,1,1,2,1,
     1,3,2,2,1,1,2,1,1,1,2,1,1,4,
     2,2,1,3,2,2,1
308  DATA 0,23,1,1,7,2,2,1,1,2,1,
     1,8,2,1,1,4,2,1,1
309  DATA 0,23,1,1,8,2,1,1,5,2,4,
     1,6,2,2,1
310  DATA 0,23,1,1,2,2,6,1,2,2,4,
     1,4,2,1,1,6,2,2,1
311  DATA 0,22,4,1,6,2,2,1,9,2,1,
     1,7,2,2,1
312  DATA 0,21,1,1,2,2,1,1,8,2,1,
     1,10,2,1,1,5,2,4,1
313  DATA 0,20,2,1,1,2,1,1,8,2,1,
     1,12,2,1,1,3,2,4,1,1,2,1,1
314  DATA 0,19,3,1,1,2,1,1,8,2,1,
     1,12,2,1,1,3,2,3,1,2,2,2,1
315  DATA 0,19,3,1,1,2,1,1,8,2,1,
     1,8,2,6,1,7,2,3,1
316  DATA 0,18,4,1,1,2,5,1,4,2,1,
     1,4,2,4,1,4,0,5,1,4,2,4,1
317  DATA 0,17,1,1,3,2,1,1,1,2,1,
     1,4,0,10,1,7,0,2,1,3,3,1,1,3
     ,2,3,1,1,2,1,1
318  DATA 0,16,1,1,6,2,1,1,5,0,2,
     1,1,0,1,1,2,0,2,1,8,0,1,1,5,
     3,1,1,2,2,2,1,2,2,1,1
319  DATA 0,16,1,1,4,2,2,1,1,2,1,
     1,7,0,1,1,11,0,1,1,4,3,1,1,2
     ,2,1,1,1,2,1,1,3,2,1,1
320  DATA 0,15,3,1,2,2,1,1,4,3,1,
     1,5,0,3,1,9,0,1,1,8,3,1,1,4,
     2,3,1
321  DATA 0,15,3,1,1,2,1,1,2,3,1,
     1,3,3,1,1,3,0,1,1,3,4,3,1,5,
     0,2,1,4,3,1,1,4,3,1,1,2,2,4,
     1
322  DATA 0,15,3,1,1,2,1,1,4,3,1,
     1,2,3,3,1,4,4,1,1,2,3,6,1,2,
     3,2,1,3,3,1,1,2,3,1,1,1,2,5,
     1
323  DATA 0,15,3,1,1,2,1,1,1,3,1,
     1,2,3,2,1,4,3,5,1,9,3,1,1,8,
     3,1,1,1,2,5,1
324  DATA 0,15,3,1,1,2,1,1,6,3,1,
     1,2,3,1,1,4,2,3,1,6,3,1,1,2,
     2,6,3,1,1,5,2,2,1
325  DATA 0,15,3,1,1,2,1,1,3,3,1,
     1,2,3,3,1,8,2,6,1,4,2,1,1,3,
     3,2,1,6,2,1,1
326  DATA 0,3,3,1,10,0,1,1,2,2,1,
     1,6,3,1,1,20,2,4,1,8,2,1,1
327  DATA  0,2,1,1,3,2,2,1,8,0,1,
     1,1,2,1,1,1,2,1,1,4,2,1,1,33
     ,2,1,1
328  DATA 0,2,1,1,5,2,2,1,6,0,4,1
     ,1,2,4,1,34,2,1,1
329  DATA 0,2,1,1,7,2,1,1,6,0,2,1
     ,28,2,1,1,4,2,3,1,4,2,1,1
330  DATA 0,2,1,1,8,2,1,1,5,0,3,1
     ,5,2,1,1,21,2,2,1,4,2,6,1
331  DATA 0,3,1,1,8,2,1,1,5,0,2,1
     ,1,2,2,1,2,2,1,1,21,2,3,1,4,
     2,5,1
332  DATA 0,4,2,1,7,2,1,1,5,0,4,1
     ,3,2,1,1,20,2,3,1,4,2,5,1
```

```
333  DATA 0,6,1,1,6,2,4,1,4,0,2,1
     ,1,2,1,1,1,2,1,1,20,2,4,1,4,
     2,3,1
334  DATA 0,6,2,1,4,2,1,1,4,2,1,1
     ,4,0,1,1,1,2,3,1,16,2,3,1,1,
     2,9,1
335  DATA 0,5,1,1,2,2,2,1,1,0,1,1
     ,6,2,1,1,4,0,4,1,19,2,6,1,1,
     2,3,1
336  DATA 0,5,1,1,4,2,2,1,6,2,1,1
     ,8,0,1,1,16,2,2,1,7,2,4,1
337  DATA 0,5,1,1,5,2,1,1,6,2,1,1
     ,7,0,2,1,25,2,4,1
338  DATA 0,6,2,1,3,2,1,1,7,2,5,1
     ,2,0,1,1,27,2,4,1
339  DATA 0,5,1,1,2,2,3,1,2,2,2,1
     ,9,2,3,1,25,2,6,1
340  DATA 0,5,1,1,4,2,1,1,3,2,1,1
     ,10,2,1,1,23,2,1,1,2,2,7,1
341  DATA 0,5,1,1,4,2,1,1,13,2,1,
     1,24,2,1,1,1,2,5,1,2,2,2,1
342  DATA 0,6,6,1,4,2,1,1,6,2,1,1
     ,25,2,1,1,2,2,3,1,3,2,3,1
343  DATA 0,12,1,1,10,2,1,1,25,2,
     1,1,2,2,2,1,4,2,4,1
344  DATA 0,12,1,1,4,2,2,1,4,2,1,
     1,24,2,1,1,3,2,1,1,5,2,5,1
345  DATA 0,13,4,1,2,0,4,1,14,2,6
     ,1,3,2,2,1,7,2,2,1,1,2,6,1
346  DATA 0,22,1,1,13,2,1,1,6,2,3
     ,1,9,2,1,1,2,2,5,1,1,2,1,1
347  DATA 0,21,1,1,14,2,1,1,19,2,
     1,1,1,2,4,1,3,2,1,1
348  DATA 0,21,1,1,14,2,1,1,18,2,
     1,1,2,2,4,1,4,2,1,1
349  DATA 0,20,1,1,15,2,1,1,2,2,1
     ,1,14,2,1,1,3,2,4,1,2,2,3,1
350  DATA 0,20,1,1,16,2,2,1,12,2,
     3,1,3,2,2,1,5,2,4,1
351  DATA 0,20,1,1,16,2,1,1,5,2,1
     ,1,7,2,1,1,1,12,2,4,1
352  DATA 0,19,1,1,17,2,1,1,4,2,1
     ,1,4,2,4,1,13,2,5,1
353  DATA 0,19,1,1,18,2,1,1,3,2,1
     ,1,4,2,1,1,15,2,6,1
354  DATA 0,19,1,1,17,2,1,1,1,2,4
     ,1,4,2,1,1,15,2,6,1
355  DATA 0,19,1,1,16,2,1,1,5,2,1
     ,1,4,2,1,1,14,2,5,1,1,2,1,1
356  DATA 0,19,1,1,16,2,2,1,4,2,1
     ,1,3,2,1,1,21,2,1,1
357  DATA 0,19,1,1,16,2,4,1,3,2,3
     ,1,17,2,5,1
358  DATA 0,19,1,1,16,2,5,1,21,2,
     5,1,2,2,6,1
359  DATA 0,19,1,1,16,2,4,1,20,2,
     14,1,1,2,1,1
360  DATA 0,19,1,1,16,2,2,1,21,2,
     4,1,3,2,3,1,1,2,4,1,1,2,2,1
361  DATA 0,19,1,1,17,2,1,1,20,2,
     4,1,4,2,2,1,3,2,2,1,1,2,4,1
362  DATA 0,19,1,1,17,2,2,1,18,2,
     7,1,3,2,1,1,3,2,1,1,2,2,4,1
363  DATA 0,19,1,1,18,2,5,1,13,2,
     5,1,2,2,2,1,3,2,1,1,8,2,1,1
364  DATA 0,19,1,1,18,2,4,1,13,2,
     5,1,3,2,2,1,2,2,1,1,7,2,2,1
365  DATA 0,20,1,1,11,2,3,1,3,2,2
     ,1,15,2,6,1,3,2,1,1,2,2,1,1,
     6,2,3,1
366  DATA 0,21,1,1,9,2,1,1,3,2,5,
     1,15,2,5,1,14,2,3,1
367  DATA 0,20,2,1,8,2,1,1,23,2,7
     ,1,13,2,3,1
368  DATA 0,19,1,1,2,2,1,1,7,2,1,
     1,23,2,6,1,15,2,1,1
369  DATA 0,19,1,1,1,2,1,1,1,2,1,
     1,6,2,1,1,3,2,1,1,19,2,8,1,1
     3,2,1,1
370  DATA 0,19,1,1,1,2,1,1,2,2,1,
     1,4,2,1,1,3,2,1,1,20,2,7,1,1
     3,2,1,1
371  DATA 0,19,1,1,2,1,1,2,2,2,
     1,3,2,1,1,2,2,1,1,5,2,1,1,16
     ,2,5,1,13,2,1,1
372  DATA 0,20,1,1,1,0,2,1,2,0,2,
     1,1,0,1,1,2,2,1,1,4,2,1,1,14
     ,2,7,1,13,2,1,1
373  DATA 0,29,1,1,2,2,1,1,4,2,1,
     1,11,2,3,1,4,0,4,1,11,2,1,1
374  DATA 0,30,1,1,2,2,1,1,2,2,2,
     1,4,2,2,1,3,2,2,1,8,0,4,1,8,
     2,2,1
375  DATA 0,31,5,1,1,0,1,1,3,2,1,
     1,2,0,3,1,11,0,3,1,6,2,2,1
376  DATA 0,38,3,1,20,0,6,1
377  DATA 0,1,0,1,0,1
378  DATA 0,5,1,5,3,0,2,5,6,0,2,5
379  DATA 0,5,1,5,3,0,2,5,1,0,2,5
     ,3,0,2,5
380  DATA 0,5,1,5,3,0,2,5,1,0,2,5
     ,3,0,2,5
381  DATA 0,5,6,5,6,0,2,5
382  DATA 0,5,1,5,3,0,2,5,1,0,2,5
     ,3,0,2,5
383  DATA 0,5,1,5,3,0,2,5,1,0,2,5
384  DATA 0,5,1,5,3,0,2,5,1,0,2,5
     ,3,0,2,5
```

麥法基專欄

掀開蘋果高解像畫面的神秘面紗

圖形繪畫法研究（二）

上一期討論的圖形繪畫法得到一個結論是假如圖形垂直長度在8個單位以內而垂直移動亦同時是8個單位，加上開始時圖形在畫面最頂部開始，則繪畫是十分容易的事。最主要是我們把圖形處理盡量符合高解像畫面的限制內，換來的好處是圖形得以最快速度進行，這一期再深入研究這種繪圖法的應用。

當然，這種繪圖方法限制甚大，在實際物體移動絕對不適用，不過我們可以試想並非所有圖形都會在畫面上移動，例如希望在高解像畫面印出字形或分數之類便是了。要在高解像畫面印出文字幕的字形，讀者可以隨便找到超過十個以上的工具程式或其他書籍內的有關文章加以利用。但往往我們存在兩個問題：我們可能只需要幾個或十個字形而已，是否應該把這些程式通通全部照搬過來？其二是如何把這些程式應用到自已作品內？

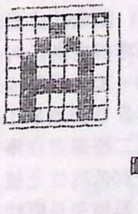
工具程式要考慮到全面性的用途，因此勢必長而且加入編輯部份讓讀者易於運用，要摘取其中一部份讓自己加以應用實在浪費時間，最直接的方法還是了解造字原理讓自己靈活運用。

請參看圖一的一個A字母的字形，該字形寬度只有7點而高度只有8點，寬度7點只佔用一個位元組（BYTE）而高度8點剛巧乎合目前討論繪圖法的條件。假如我們把整個字形加以解剖以BYTE來顯示，則會成爲下面的數值：

綫	16進制	2進制							
第一條	00	0	0	0	0	0	0	0	0
第二條	08	0	0	0	0	1	0	0	0
第三條	14	0	0	0	1	0	1	0	0
第四條	22	0	0	1	0	0	0	1	0
第五條	22	0	0	1	0	0	0	1	0
第六條	3E	0	0	1	1	1	1	1	0
第七條	22	0	0	1	0	0	0	1	0
第八條	22	0	0	1	0	0	0	1	0

四方形以外的零是判斷顏色的高位元，由於字形只用一個位元組，所以顏色全部暫時無須理會，以零存入最高位元。如果我現時按HGR清除畫面，再按CALL－151 進入監督部份，照著以下步驟把這形16進數值存入，則畫面左上角便會有一個A字母圖形顯示出來。

```
2000 :00
2400 :08
2800 :14
2C00 :22
3000 :22
3400 :3E
3800 :22
3C00 :22
```



圖一

現在重溫一下上一期的流程圖，原則上和畫取一個垂直長方形沒有甚麽分別，問題是在取圖形數值那一項。繪畫長方形所有數值是＄FF，所以上一期的原始碼內我們只用一個LDA ＃＄FF 指令便可以完成工作，目前每一條綫的數值都有所不同，程序上需有所改變了。

我們剛才鍵入的數值，事實上是從已經造好的位元組圖表按次序逐一輸入。換言之，這些16進的數值，便是一個圖形表（SHAPE TABLE）。圖形表必須存放在電腦記憶體內，讓電腦在繪畫過程中取用。我們可以把它任意存放到RAM ADDRESS 內，只要這些地方不會和電腦其他用途有所衝突便可。請把上一期的原始碼重新輸入MERLIN ASSEMBLER ，改爲一個繪畫圖形的程式。BOOT 起MERLIN 後，只要按L然後輸入檔名，整個原始碼便重新顯示在EDITOR 內。

LISTING ＃1 是一個已經修改成爲一個在畫面左上角繪畫出一個A字母的原始程式。讀者祇要把上一期的程式加入新的部份便成。我們研究一下新加入部份指令，第7和第8的SHPL和SHPH 是指向圖形表的零頁指位器。倘若我們把圖形表存放的開始地址以低高位元分別存入該兩個零頁位置，程式開始繪畫圖形會到指向的地方找尋圖形資料。問題是如果超過一個以上圖形時，當圖形完畢後；指位器的數值亦要改爲指向另外一個圖形位址。因此在未繪畫圖形前，首先要把指位器的數值交待清楚。第15行的WORDSETUP 便是把圖形表位址分別輸入SHPH和SHPL 。目前我們的圖形資料存放在尾部WORDA 處，那是由＄6037開始，換言之，SHPH的資料數值必須是＄60而SHPL的數值必須是＄37。我們無須費時去找尋這些數值，只要在圖形表的標籤欄內加上一個記號，（這裏用的是WORDA） 讓組合器自己去找尋。其實這樣已經比BASIC 語言更加便利，因爲在BASIC 檔中，我們必須要加上行數才能應用GOSUB 或GOTO 之類指令。另一個理由是當我們不斷修改、編寫或加上其他部份後，圖形表的地址隨時隨地會被改變，尋找地址的工作還是交由組合器去負責吧。第15行你會看到一個LDA ＃＞WORDA指令，這是把WORDA 的高位元值（＄6037的＄60）存入累加器內，再經由STA SHPH 從累加器送到 SHPH 指位器。同理跟着的LDA ＃＜WORDA亦把WORDA的低位元值（＄6037的＄37）轉送到SHPL。記着任何的資料轉移必須由LDA和STA 兩個指令負責，所以在機器語內這兩個指令出現次數最多。

原則上只要把指位器值加以設定後便可以立即進行繪圖工作了。不過爲了以後可能會有超過一個以上圖形，我們加入RTS把WORDSETUP 變爲一個副程式，好處是把它和主程式分割開，當要畫出A字時去呼叫應用，不用時便可以置諸不理。

繪圖部份亦要作出適當修改，上一次我們繪畫長方形只要把＄FF存入畫面位址便可，現在要存入圖形表資料，我們要改用第32行的LDA（SHPL，X） 指令。相信我們會有一個不明白的地方，爲甚麽這裏用LDA（SHPL，X） 而下一行却用STA（HIRESL），Y這麽麻煩。兩個指令究竟有些甚麽分別？甚麽時候用LDA（SHPL，X） 而甚麽時候用LDA（HIRESL），Y？SHPL和HIRESL 不都是零頁指位器嗎？

```
: A S M
                  1            ORG   $6000
6000: 4C 30 60    2            JMP   DRAWWORD
                  3   *
                  4   *
                  5   HIRESL   EQU   $26
                  6   HIRESH   EQU   $27
                  7   SHPL     EQU   $50
                  8   SHPH     EQU   $51
                  9   *
                  10  *
                  11  *
                  12  YVERTH   EQU   $9200
                  13  YVERTL   EQU   $92C0
                  14  *
6003: A9 60       15  WORDSETUPLDA   #>WORDA
6005: 85 51       16           STA   SHPH
6007: A9 37       17           LDA   #<WORDA
6009: 85 50       18           STA   SHPL
600B: 60          19           RTS
                  20  *
                  21  *
600C: B9 C0 92    22  GETADR   LDA   YVERTL,Y
600F: 85 26       23           STA   HIRESL
6011: B9 00 92    24           LDA   YVERTH,Y
6014: 85 27       25           STA   HIRESH
6016: 60          26           RTS
                  27  *
                  28  *
6017: A0 00       29  DRAW     LDY   #$00
6019: 20 0C 60    30           JSR   GETADR
601C: A2 00       31           LDX   #$00
601E: A1 50       32  DRAW2    LDA   (SHPL,X)
6020: 91 26       33           STA   (HIRESL),Y
6022: A5 27       34           LDA   HIRESH
6024: 18          35           CLC
6025: 69 04       36           ADC   #$04
6027: 85 27       37           STA   HIRESH
6029: E6 50       38           INC   SHPL
602B: C9 40       39           CMP   #$40
602D: 90 EF       40           BCC   DRAW2
602F: 60          41           RTS
                  42  *
                  43  *
6030: 20 03 60    44  DRAWWORD JSR   WORDSETUP
6033: 20 17 60    45           JSR   DRAW
6036: 60          46           RTS
                  47  *
                  48  *
                  49  *
6037: 00 08 14
603A: 22 22 3E
603D: 22 22       50  WORDA    HEX   0008142223E2222
                  51  *

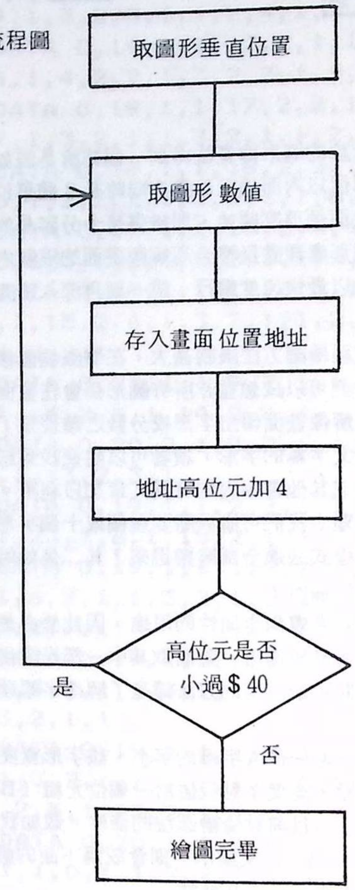
--End assembly--

63 bytes

Errors:: 0
```

列表一

圖二 流程圖



6502 的機器語指令有 3 個以供查表之用，分別如下：

㈠ LDA AAAA，X
　 LDA AAAA，Y

㈡ LDA（AA），Y

㈢ LDA（AA，X）

四個A表示記憶體地址，兩個A表示零頁指位器。而LDA或STA都一樣，第一種指令無論用X或Y暫存器都可以作爲索引。第二種只能用Y作爲索引而第三種只能用X作爲索引。

第一種最簡單容易明白，意思是把AAAA地址加上X或Y的值便是我們要找尋的地址，例如我們已知＄7000是WORDA 開始的位置而它總共有 8 個不同資料至＄7007 爲止。要尋找內裏任何一個資料只要輸入X或Y值然後LDA，＄7000，X 便可以找到。X 的數值由 0 至 7 都可以作爲索引去找尋＄7000至＄7007 之間資料。

第二種筆者在第15期亦曾加以解釋，電腦首先到零頁指位器找到低高位元值，把它重新結合成爲開始位置，再加上Y暫存器值便是要找尋的地址。和第一種不同的地方是第一

種的WORDA（開始位址）立即被利用作爲査表之用，而第二種要通過零頁指位器去找出來。

至於第三種比較令人難於明白。LDA（AA，X） 內的X暫存器是用來找尋零頁區指位器之用。它不會直接參與找尋所需的資料。倘若指令是LDA（$2A，X）而X的值是1的話，該指令會首先把$2A 加上1，於是指位器會被更改到$2B（所更改以低位元指位器爲對象），原來的指位器是$2A和$29會變成$2B和$2A，所以要找尋資料的正確地址便由$2B和$2A去組合而成。換言之，如果我們不希望指位器位置改變的話，只要把X暫存器永遠輸入0值才能應用。那麽怎樣去找尋資料呢？很簡單，把低位元或高位元指位器的內在值改變便成了。

現在看看第31行我們輸入DX $00，這個X值我們並不加以改變，而SHPL的値在 WORDSETUP 副程式內已經被設定爲$37，SHPH被設定爲$60，因此LDA（SHPL，X）便會跑到$6037的WORDA 開始位址去找尋資料。再經由STA（HIRESL），Y 指令搬到畫面位址上由34至37行把畫面垂直座標增加一綫，而第38的INC SHPL 便是把SHPL 增加一個數値，當程式跑到32行的LDA（SHPL，X）時，該指令會由$6038去提取資料，如此直至整個A字母被繪畫出來爲止。

一部份讀者或者已經察覺到這個程式的LDA（SHPL，X）倘若改爲LDA WORDA，X而第38行的INC SHPL改爲INX，則程式仍然可以照樣繪畫出A字母而絲毫沒有錯誤。爲甚麽還要利用LDA（SHPL，X）這個複雜指令？事實上效果的確是一樣，不果倘若WORDA 是一個複雜而巨大的圖形時，LDA WORDA，X 便不能被應用，因爲X或Y值都不能超過$FF。又或者假如WORDA 是處於$60FF 的開始位址時，使用LDA WORDA，X當X値被加到$FF時會變爲$00，換言之圖形地址不會是$6100而是$6000，這樣便會造成錯誤的査表。而使用LDA（SHPL，X）只要同時改變SHPH 便可以正確找到位置了。

要看看程式效果，只要把PIL存入電腦，然後HGR再CALL 24576 便可。■

LODE RUNNER MAKER

米酒

LODE RUNNER這個精彩遊戲，相信讀者一定玩過，當推出第二集筆者便立即買了回家玩過痛快，誰知每版都是陷阱重重，難度高得很！盡了最大努力也祇能玩至5個LEVEL，相信許多讀者也曾有相同經驗吧！

後來由朋友告知（CTRL－L）的資料是放在磁軌$C，磁區$D（電腦時代十八期也曾刊出），祇要適當更改該磁區資料便能玩更高LEVEL，但問題也隨之而來，首先我們必須使用SE－CTOR EDITOR 這類工具，再以十六進數更改適當位址內容才能達到目的，對資深用家來說，出錯機會不大，但對普通用家，連SECTOR EDITOR也未必識得應用，再要以十六進數更改內容，相信出錯機會很大，事實上筆者也曾出錯，使磁碟DOWN 掉，祇出版頭，不能開始。

因此筆者認爲這些十六進數、位址更改用程式做是比人手更快、更理想，相信讀者也有同感，因此筆者寫了這個程式來應用，也希望讀者應用時減少出錯。

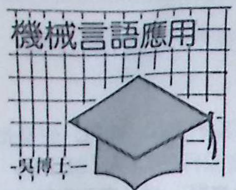
這個程式是利用DOS的RWTS 副程式，對於那些對DOS有研究的讀者，RWTS 是耳熟能詳了！這個程式祇要略作更改，也可應用它來更改其他的GAMES！

讀者祇要RUN LODE RUNNER MAKER便可應用它，程式會指示讀者各步驟，先放入LODE RUNNER Ⅱ磁碟然後按鍵，再鍵入你的名字及RE－TURN。最後是鍵入LEVEL 及人數。如果資料有錯誤可按N，否則按Y便大功告成。

最後一提是讀者可試玩LEVEL 50，如果有方法完成便可出一密碼，配合原廠說明書便可寄回LODE RUNNER的軟件公司取證書一張！證明你是LO－DE RUNNER 畢業生呢！■

```
1  REM  *********************
2  REM  * LODE RUNNER MAKER *
3  REM  *     BY  MIKE      *
4  REM  *     10-8-1985     *
5  REM  *********************
10  FOR I = 4096 TO 4110: READ A:
     POKE I,A: NEXT I
20  POKE 47083,0: POKE 47084,12: POKE
    47085,13: POKE 47088,0: POKE
    47089,32: POKE 47092,1
30  DATA  32,227,3,32,217,3,144,2
    ,133,0,169,0,133,72,96
50  TEXT : HOME : SPEED= 220
60  FOR X = 1 TO 40 STEP 1:Y = 1:
     GOSUB 1000: NEXT X
70  FOR Y = 1 TO 22:X = 40: GOSUB
    1000: NEXT Y
80  FOR X = 40 TO 1 STEP  - 1:Y =
    22: GOSUB 1000: NEXT X
90  FOR Y = 22 TO 1 STEP  - 1:X =
    1: GOSUB 1000: NEXT Y
100  POKE 32,1: POKE 33,38: POKE
    34,1: POKE 35,20: HOME
110  PRINT "WELLCOME TO THE LODE
    RUNNER MAKER!"
120  PRINT "THIS PROGRAME IS WRIT
    E BY MIKE CHAU"
125  PRINT : HTAB 12: INVERSE : PRINT
    "COMPUTER AGE": NORMAL
130  PRINT : PRINT "PLEASE TAKE T
    HE LODE RUNNER DISK INTO DRI
    VE 1 ,AND PRESS ANY KEY": GET
    A$
140  CALL 4096
200 B$ = "        "
210  HOME : PRINT "OK!": INPUT "W
    HAT NAME WOULD YOU LIKE ?(1-
    8 WORDS)";A$
220  IF  LEN (A$) = 8 THEN 240
225 A$ = A$ +  RIGHT$ (B$,8 -  LEN
    (A$)): GOTO 240
230  GOTO 210
240  FOR I = 1 TO 8: POKE 8191 +
    I, ASC ( MID$ (A$,I,1)) + 12
    8: NEXT I
250  INPUT "WHAT LEVEL WOULD YOU
    LIKE?(1-50)";B: IF B < 1 THEN
    250
260  IF B > 50 THEN 250
270  POKE 8200,B: POKE 8201,B - 1

290  INPUT "WHAT LIVES WOULD YOU
    LIKE?(0-255)";C: IF C < 0 THEN
    290
300  IF C > 255 THEN 290
310  POKE 8202,C
320  PRINT "ALL DATA IS SATISFIED
     ?": GET D$
330  IF D$ = "Y" THEN 350
340  RUN
350  POKE 47092,2
360  CALL 4096
370  HOME : PRINT "OK! ALL DATA P
    UT IN LODE RUNNER DISK": PRINT
    "PRESS ANY KEY TO PLAY!": GET
    A$
380  PR# 6
1000  HTAB X: VTAB Y: PRINT "/"; CHR$
    (8);"-"; CHR$ (8);"*": RETURN
```



眞正隨機數的產生

在APPLESOFT的MANUAL 中曾經列出過一個用作產生隨機數（RANDOM NUMBER ）的指令——RND，這個指令在很多遊戲及程式中都會用到，本來是好像沒有甚麼好談的，不過這個指令並不是每一次所產生的數字都會不同，較直接說，其實它每次所產生的都是一樣。這樣一來這個APPLESOFT的指令就有著一個BUG了，若果你的BASIC遊戲程式用到這個指令的話，你會發現每次啓動時所見到的螢幕都會相同。

以上的情形最簡單的試驗是先把電腦關掉，然後啓動電源，按下RESET鍵，鍵入PRINT RND (1)，電腦便會顯示出0・973136996。現在你把電源關掉，再重複一次以上的步驟，你會發現到電腦顯示的仍然是同一個數字。

爲了解決以上這個問題，我從MANUAL 中第141頁找到幾個記憶體位置，由＄C9到＄CD共5個零頁指標，這5個位置是APPLESOFT用來取得隨機數的，以上的BOG亦源出於此。

不過APPLE亦有一個眞正的隨機數產生器，它是由＄4E至＄4F（十進制是78及79），這兩個位置可以被用來改善RND指令，方法是用以下的機械語言來達到目的，這個程式可以隨意移位，現在我假設由＄300開始。

```
300:AD 10 C0 20 0C FD A6 4E
    86 CA A6 4F 86 CB 60
```

```
300：AD  10  C0   LDA  $C010；清除鍵盤
303：20  0C  FD   JSR  $FD0C；置定$4
                       E及$4F，並等候按鍵
306：A6  4E       LDX  $4E；將$4E中數
                       值載入 X暫存器
308：86  CA       STX  $CA；將X暫存器內
                       容存到$CA中 即APPLES
                       —OFT 的隨機數產生器中
30A：A6  4F       LDX  $4F；將$4F數值
                       載入X暫存器（另一個隨機數）
30C：86  CB       STX  $CB；將X暫存器內
                       容存入$CB，另一個APPLE
                       —SOFT隨機數產生器位置。
30E：60           RTS  ；程式結束
```

現在你可以將這個程式貯存到磁碟上，按入

BSAVE RND A$300，L$10

這時你可以將它加入到其它的程式上去，只要在程式開始時，先將這個檔案BLOAD入記憶體中，然後以CALL 768來啓動程式，在以後的RND指令都會產生一個眞正的隨機數了，你不妨將這程式加入到你昔日的BASIC程式中，我可以保證程式每次所出示的東西都會不同。■

配合80字行咭使用的

兩欄目錄檔案名字顯示

本程式是一個簡單而又實用兩欄目錄檔案名顯示，需要80字行咭配合才能發揮程式的優點。大家只要BRUN DOU—BLE COLUMN CATALOG，在80字行顯示時，檔案名會分兩欄顯示，而在40字行顯示時，會回復一欄顯示。在兩欄顯示時　，大家可如常利用ESC I，J，K，M 來控制游標。

假如大家想有永久的兩欄目錄檔案名顯示功能，先BR—UN DOUBLE COLUMN CATALOG，此時DOS 會被更改，然後INIT一張新磁碟。

本文程式只有一個，大家可參照程式列表打入，然後用BSAVE DOUBLE COLUMN CATALOG，A$300，L$78存入磁碟，或在即期的程式磁碟內找到相同名字的FI—LE。

打入機械碼方法可參閱第十七期「如何輸入機械碼」。

```
*****************
* DOUBLE COLUMN *
*    CATALOG    *
*  A$300,L$78   *
*****************
0300- A2 00 BD 26 03 F0 1E 85
0308- 86 E8 BD 26 03 85 85 E8
0310- BD 26 03 85 87 E8 A0 00
0318- BD 26 03 91 85 E8 C8 C6
0320- 87 D0 F5 F0 DD 60 AD A8
0328- 06 EA A9 8D 20 ED FD AD
0330- C3 06 EA A9 8D 20 ED FD
0338- AD EB 30 A0 07 0A 30 03
0340- 88 D0 FA B9 A7 B3 20 ED
0348- FD A9 A0 20 ED FD BD E7
0350- B4 85 44 BD E8 B4 85 45
0358- 20 42 AE A9 A0 20 ED FD
0360- A0 1D BD C9 B4 C9 A0 B0
0368- 02 A9 AA AE 21 01 F0 AE
0370- 2F 05 A2 03 20 4A F9 00
0378- FF
```
■

麥法基 遊戲專欄

GRAPHICS MAGICIAN編寫的遊戲：

立體死亡大賽車

筆者6個月前已經計劃編寫一個立體賽車遊戲，但一直無法實現。立體賽車在蘋果電腦較爲出色的有SPEEDWAY CLASSIC和一個以低解像度編寫的富士山大賽車；至於FORMULA I 和一個PITSTOP Ⅱ顯然不甚理想。事實上，要在蘋果電腦產生逼眞的立體賽車效果並非容易。就算如SPEEDWAY CLASSIC 亦感覺上好像缺少了點甚麽，最主要是動感和刺激性還有待增強。

最初筆者著手以一頁高解像畫面去構思整個遊戲，但無法解決背景問題。一個高速賽車的跑道必須有飛快的奔馳效果，而且和平面賽車有所不同的地方是要照顧向左右轉彎，當然跑道亦不可能以圖形表處理，因爲要建立差不多半個螢幕的圖形表十分費時。

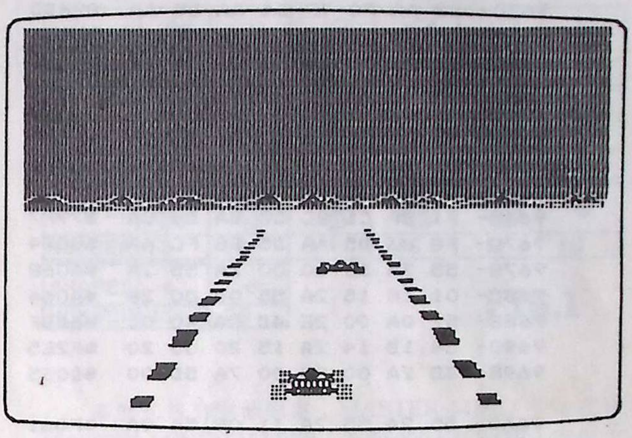
更加複雜的是背景上的遠山，當跑道向左右旋轉時遠山背景亦要同時以相反方向移動。倘若遠山背景是白色的話則較爲易辦，移動方法只要把該部份圖形資料以一個位元組爲單位向左右作整個畫面旋轉便可。但遠山背景並非全部是白色，而且天空是藍色，這樣移動必須爲兩個位元組作爲單位，但如此移動效果會變得不自然而且有跳動的感覺。

最後筆者發覺單靠一個畫面無法解決所有問題。換言之，遊戲必須以兩頁交替的PAGE FLIPPING 進行。首先跑道會有飛馳效果，亦同時消除閃動的物體。原則上用BLOCK的繪圖法亦可以解決物體閃動問題，但BLOCK 的繪圖法在兩頁運用上不靈活，因此採用XDRAW 繪圖法。其次是跑道的畫面，跑道由Y＝107 至Y＝191 伸展，每個畫面有左右各一條，由第一頁和第二頁畫面交替出現，所以每次實際有4條跑道圖形不斷在玩者眼前交替顯示。加上左右旋轉共有12條跑道圖形需要加以處理，因此遊戲需要把DOS 移上RAM CARD ，讓較多的記憶去處理圖形。

遊戲中跑道圖形表並非以一般圖形表方法建立，爲了簡化起見，地面全部以黑色作爲背景，而跑道是白色。圖形表以三個位元組作爲一組表示，第一個位元組存入垂直Y，第二個存入水平位元組值而第三個是圖形資料值，如此一個跑道圖形資料大約只用300個BYTE左右去儲存，所以移去DOS後多出的10K 記憶可以應付綽綽有餘。

遊戲以玩者立場去編寫，所以盡量造到快速以富有刺激性。是綜合幾個賽車遊戲的長處和特色加以改進而成。至於遠山的移動分雙單兩組圖形資料交替在兩頁上移動，這樣旋轉時較爲圓滑。

讀者請把LISTING #1 PIC ADR 名稱存放到第22期磁碟上才運行遊戲，檔案由＄9600開始，長度爲＄1F0，記着要把DOS 移上RAM CARD 後才鍵入LISTING 和BSAVE。



兩頁交替的動畫遊戲適合對背景有較多要求的程式，但編寫程序方面很易出錯，因爲當玩者觀看第一頁畫面時，所有繪圖及程式運行都在第二頁工作。換言之，玩者所看見的畫面都是已經運行完的效果。以繪上新圖形而抹去舊圖形的原理去推斷，則我們所看到的是新圖形，而舊圖形存在第二頁上。所以在設計動畫時，要留意舊圖形並非和新圖形同時存在一頁，尤其是加上圖形變換會更加複雜。雖然在程式設計上要應付多方面的問題，然而大部份職業水準的遊戲却樂於採用，因爲可創出甚佳的視覺效果。

要編寫兩頁交替的動畫首先必須要對單一頁繪圖有充份瞭解，否則編寫過程中DEBUG甚爲困難，筆者有機會會在專欄中和大家深入研究，下次再會。

編者註：本遊戲是由多個FILE 組成，除了PIC ADR（見列表一）外，全部都收錄在第22期的程式磁碟上，程式磁碟可在各特約經銷處購得。在玩這遊戲前，大家必須鍵入LISTING 1 ，存放在同一磁碟上才可運行。必須要緊記的是，在鍵入列表一前，必須要把DOS 移上RAM咭（搬移DOS 的辦法可參閱電腦時代第十期〔將DOS 移上RAM 咭〕一文），否則後果恕不負責。同時，爲減少大家在輸入程式後偵錯的不便，我們在程式列表後加付上一個CHECKSUM TABLE，大家在鍵入完列表及BSAVE 入碟後，可在第22期程式磁碟上執行一個CHECKSUM TABLE，然後再LOAD入PIC ADR 校對一次CHECK SUM，如有錯，可利用第十五期P54上刊出的〔機械碼輸入編輯器〕編輯。

由於本遊戲需要較大記憶空間，故大家必須擁有一張16K或以上的RAM咭。

電腦時代長期訂戶收到的程式磁碟是錄齊所有FILE ，故無須鍵入任何列表，只須直接RUN 遊戲即可。■

立體死亡大賽車

```
******************
*    PIC ADR     *
*  A$9600,L$1F0  *
******************
9600- 85 87 D0 AA D5 AA F1 A8   $A1C6
9608- D5 AA 85 80 D5 AA D5 AA   $53F7
9610- F1 A9 D5 E2 D4 AA D5 AA   $0055
9618- D5 A0 D5 8A D5 AA D5 AA   $A010
9620- E5 AA D5 AA D5 8A D5 AA   $3850
9628- F1 BF C3 AA D5 AA FC A3   $EF13
9630- D5 AA F0 9F D4 AA D5 AA   $26DD
9638- FC A3 D5 F2 91 A8 D5 AA   $1923
9640- 85 A6 D5 E0 D0 AA D5 AA   $A951
9648- F5 A8 D5 AA D5 B2 99 AA   $37B0

9650- FC FF 8D AA D5 80 B3 86   $15B0
9658- D4 82 F3 FC 81 AA D5 82   $DA10
9660- F7 A6 D5 98 93 A3 95 AA   $33C5
9668- F1 8F C1 8C C3 8A D4 8A   $79F7
9670- F8 83 D5 AA D5 B8 FC AA   $80F4
9678- 55 3A 55 00 00 2A 55 2A   $A068
9680- 01 18 15 2A 55 00 00 28   $A064
9688- 57 0A 00 2E 4D 0A 40 00   $6BBF
9690- 34 1B 14 2A 15 20 05 20   $F2E5
9698- 3D 2A 00 00 00 7A 5D 00   $D035

96A0- 55 2A 55 2A 11 08 54 0A   $F061
96A8- 50 2A 55 2A 55 0A 51 2A   $5824
96B0- 55 2A 45 6B 15 2A 51 22   $27CC
96B8- 55 3A 55 20 55 28 50 2A   $BD55
96C0- 55 2A 45 20 55 4A 55 0B   $AE64
96C8- 55 2A 55 2A 55 2A 01 22   $C464
96D0- 55 2A 55 2A 55 2A 05 02   $A470
96D8- 01 08 50 2A 55 00 54 2A   $C430
96E0- 55 2A 54 0A 54 02 55 2A   $AE24
96E8- 55 2A 55 0A 41 08 00 08   $C665
96F0- 8A 8E A0 D5 AA D5 E2 D1   $C22A
96F8- AA D5 8A 80 AA D5 AA D5   $7320
9700- E2 D3 AA C5 A9 D5 AA D5   $6AD6
9708- AA C1 AA 95 AA D5 AA D5   $0A9C
9710- CA D5 AA D5 AA 95 AA D5   $1ADC
9718- E2 FF 86 D5 AA D5 F8 C7   $B78E
9720- AA D5 E0 BF A8 D5 AA D5   $ED1D
9728- F8 C7 AA E5 A3 D0 AA D5   $FDA1
9730- 8A CC AA C1 A1 D5 AA D5   $B110
9738- EA D1 AA D5 AA E5 B2 D4   $3C72

9740- F8 FF 9B D4 AA 81 E6 8C   $9DB8
9748- A8 85 E6 F9 83 D4 AA 85   $2F9D
9750- EE CD AA B1 A6 C6 AA D4   $0BA4
9758- E2 9F 82 99 86 95 A8 95   $2C35
9760- F0 87 AA D5 AA F1 F8 D5   $4C74
9768- 2A 75 2A 01 00 54 2A 55   $8C8C
9770- 02 30 2A 54 2A 01 00 50   $487E
9778- 2E 15 00 5C 1A 15 00 01   $1FA8
9780- 68 36 28 54 2A 40 0A 40   $DB6B
9788- 7A 54 00 00 00 74 3B 01   $3CF8

9790- 2A 55 2A 55 22 10 28 15   $F8F2
9798- 20 55 2A 55 2A 15 22 55   $795A
97A0- 2A 55 0A 57 2B 54 22 45   $92E8
97A8- 2A 75 2A 41 2A 51 20 55   $67DA
97B0- 2A 55 0A 41 2A 15 2B 17   $D378
97B8- 2A 55 2A 55 2A 55 02 44   $7678
97C0- 2A 55 2A 55 2A 55 0A 04   $97D0
97C8- 02 10 20 55 2A 01 28 55   $62FA
97D0- 2A 55 28 15 28 05 2A 55   $86F8
97D8- 2A 55 2A 15 02 11 00 10   $36FA

97E0- 00 00 00 00 00 00 00 00   $D6DA
97E8- 00 00 00 00 00 00 00 00   $36FA
97F0- 00                        $96B6
```

MOCKING BOARD用家協會成立

M B C

麥法基先生每期都爲電腦時代編寫創作精彩的遊戲軟件，這些軟件深受廣大讀者的歡迎。麥先生所編寫的遊戲，除了有普通版外，還加插有MOCKING BOARD VERSION，使遊戲更生色不少。相信擁有MOCKING BOARD 的讀者都會同意。

由於麥先生編寫的遊戲作品受到廣大讀者歡迎，我們決定成立一個 MOCKING BOARD 用家軟件協會，簡稱MBC，今後麥 先生或其他作者編寫創作的MOCKING BOARD 版本遊戲只直接售與MBC 會員。當然這個MBC 會還會搜集坊間或 外國最新有關MOCKING BOARD 軟件的資料或軟件供會 員參閱或使用。此外，如有任何最新作品，會直接通知會員 ，我們不會公開發售。

■ 參加MBC 會，入會費每年60元。每個MOCKING BOARD 版本的遊戲定價十五元（連磁碟），或五元（自己出磁碟）。

參加MBC 會請親臨電腦時代讀者服務部辦理或電 3-312007 查詢。

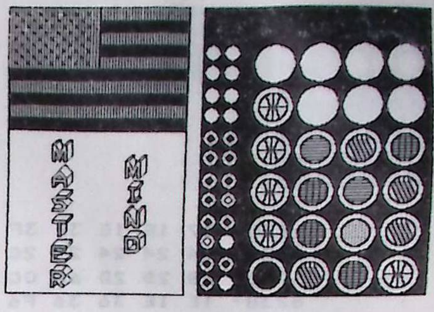
■ 凡是電腦時代COMPUSOFT磁碟月刊、微電腦技術長期訂戶及AFC會員均可直接購買各個MOCKING BOARD 版本遊戲作品，每個十五元（連磁碟）。購買請到電腦時代讀者服務部或利用郵購。

■ 麥法基先生的作品STAR WAR亦有MOCKING BOARD 版本，每張只售十五元，軟件只售與MBC 會員或長期訂戶或AFC 會員。

購買請親臨電腦時代讀者服務部或利用郵購。郵購辦法可參閱「郵購需知」。

智力棋的考驗

陳文緯



MASTER MIND 這一個遊戲相信對大家都不會陌生。以下爲你介紹的 MASTER MIND 是由電腦判斷顏色的，可供一人玩，和以前的棋類遊戲有異。

和上幾次一樣，電腦會清除螢幕，劃出 MASTER MIND 字等，並奏出一首輕快的日本民歌。片刻後，你會看見棋盤。

玩者可以用箭咀鍵選擇圖案，然後以按 RETURN 鍵結束選擇，再用同樣的方法來選擇第二、三和四個圖案。如果發覺上一個圖案選擇錯誤的話，可以按 ECS 鍵回到上一個圖案裏，便可作更改之用途（爲什麼不用顏色，而要用圖案呢？我也曾試過用顏色的，但恐怕用 GREEN MON 的朋友分不清紅、藍和綠色，所以改用圖案代表，使用 GREEN MON 的朋友都可分享）。

當選擇完四個圖案後，電腦會在左邊給出提示。方法和普通的 MASTER MIND 相同。白色是表示圖案對但位置不對，黑色（沒有塗的圓圈）表示圖案及位置都正確。如果不是全黑的話，你就要猜第二次了。一共有六次機會。如果在六次後還猜不中，電腦繪出正確的答案，並要求你按一鍵，跟着問你是否再玩。雖然這個遊戲只供一人玩，但有兩個方法可以改成二人玩的：

一、每人輪着地猜

二、將第八十行改變：

80 A（1）＝n：A（2）＝n：A（3）＝n：A（4）＝n

這樣玩者一可以隨意改變n的數，即代替電腦的 RANDOW 圖案的數目和位置。玩者二可以猜玩者一選擇的圖案。n可以由一至六任何一個數。不過，這四個數都不可重復的。

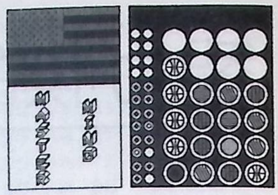
SHAPES	M	A	S	T	E	R
NUMBER	1	2	3	4	5	6
SHAPES	I	N	D			
NUMBER	7	8	9	10	11	12
SHAPES						
NUMBER	13	14	15	16	17	18
SHAPES						
NUMBER	19	20				

FIG.1

本程式共有兩個檔案：MASTER MIND 主程式和 MASTER MIND SHAPES（A＄6000，L＄63F）副程式圖形表，圖形表是以 PIXIT 繪的，內容可參看 FIG. 1。以下就是主程式的行數分析：

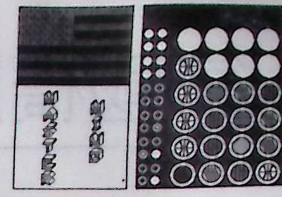
10-20	載入圖形表
30-50	劃出左邊所有物件
60	播出音樂
70	劃出棋盤
80	判斷棋子次序，數目
90-130	玩者輸入棋子和分析各鍵
140-170	檢查棋子，以給出提示
180-210	玩完的輸出
220	劃出正確的棋子
230	劃出提示棋子和音樂副程式
240	音高、音長資料

```
*************************
* MASTER MIND SHAPES *
*    A$6000,L$63F     *
*************************
6000- 14 00 2A 00 70 00 AD 00
6008- EA 00 1C 01 63 01 A1 01
6010- D3 01 13 02 54 02 59 02
6018- 54 03 EE 03 50 04 AC 04
6020- 04 05 76 05 FE 05 23 06
6028- 34 06 49 29 35 FF 1E 1E
6030- 3E 36 36 36 36 36 2E 25
6038- 24 24 24 95 B6 62 A4 52
6040- 66 24 0E 25 3C 2C 24 95
6048- 92 1A 2E 25 24 24 24 24
6050- 24 3F 37 BE F6 26 18 E4
6058- E4 0C 0C 0D 0E 0D 0C 0C
6060- 0C 2D 3E 1E 1E 45 31 36
6068- 36 36 36 1E 1E 1E 04 00
6070- 49 49 49 1E 1E 1E 1E 1E
6078- 1E 1E 1E 1E 36 36 36 2D
6080- 2C 1C 2C 35 0C 2D 35 36
6088- 2D 2C 45 2B 28 20 24 24
6090- E4 1C 1C 1C DE 9B 12 0E
6098- 0E 0E 0E 36 36 DC 40 3B
60A0- 3F 3F 67 0C 0C 15 15 0D
60A8- 48 0C 0C 04 00 49 49 29
60B0- 2D 2D 2D 36 1E 1E 1E 37
60B8- 3F 3F 3F 1E 2E 2D AD 15
60C0- 15 66 0C 0C 0C 1C 1C 96
60C8- 92 DB BA 1E 3F 3F 3F 27
60D0- 2C 2D 2D 65 E4 3F 3F F6
60D8- 0C C0 1C 64 0C 0C 25 45
60E0- 2B B0 1A 2D 2D 65 0C 0C
60E8- 04 00 49 29 2D 2D 2D 2D
60F0- 35 F7 0D 1E 1F 1E 0D 1E
60F8- 37 3F 3F 36 36 36 36 66
6100- 45 2B 28 20 24 F4 9B 92
6108- 93 1A 27 24 24 24 24 3F
6110- 3F 24 25 45 2B B0 13 2D
6118- 2D 2D 25 00 49 29 2D 2D
6120- 2D 2D 35 AF 1F 0E 1F 0E
6128- 1F 3F 3F 3F 3F 3F 0C 0C
6130- 45 B3 D2 1B 36 36 36 36
6138- 36 2D 2D 2D 2D 2D 2C 45
6140- 2B 28 20 27 FD 16 17 1E
6148- 3F 3F 3F 3F 2C 1C 2C 35
6150- 0C 2D 2C 2D E5 FB 3F 3F
6158- 27 25 6F 2D 2D 6E 3C 45
6160- 2B 04 00 49 29 2D 2D 2D
6168- 15 15 15 F6 1E 1E 0E 0E
6170- 0E 1E 1E 1E 1E 37 3F 1C
6178- 1C 27 3C 37 36 F5 3F 24
6180- 24 24 24 24 64 0C 0C 4D
6188- 49 F1 1E 1E 3F 3F 3F 8E
6190- 36 2E 2D 2D 20 1C 3F 6F
6198- 49 70 0E F6 1E 1E 76 04
61A0- 00 49 49 2D 2D 2D 3E 15
61A8- 1E 1E 36 36 36 2E F6 1E
61B0- 37 3F 3F 3F 24 25 35 44
61B8- 23 24 24 3F 24 25 0C 0C
61C0- D6 2A 2D 65 0C 9E 3A 3E
61C8- 37 36 36 36 2D 0C 44 BB
61D0- 17 04 00 49 29 35 FF 1E
61D8- 1E 3E 36 36 36 36 36 2E
61E0- 25 24 24 24 56 0E 36 1E
61E8- 4C 15 2E 2D 25 24 24 24
61F0- 24 24 3F 36 36 36 36 E6
61F8- 1C E4 1C E4 1C 2C 60 4D
6200- 15 65 0C 0C 2D 3E 1E 1E
6208- 4C 31 36 36 36 36 1E 1E
6210- 1E 04 00 49 29 2D 2D 2D
6218- 75 0E 36 36 36 F6 1E 1E
```



```
6220- 37 1E 1E 3F 3F 3F 3F 24
6228- 24 24 24 24 2C 64 0C 96
6230- 2B 2D 2D 65 0C 0C 95 F1
6238- 1E 1E 36 36 F6 D3 3F 3F
6240- 27 24 24 24 2D 2D 2D 08
6248- 9E 36 36 FE DB 0C 0C 0C
6250- 24 95 04 00 31 37 4C 26
6258- 00 49 49 49 29 2D 35 2D
6260- 35 2D 2E 2E 2E 36 35 36
6268- 35 36 3E 36 3E 36 37 37
6270- 37 3F 3E 3F 3E 3F 27 3F
6278- 27 3F 3C 3C 3C 24 27 24
6280- 27 24 2C 24 2C 24 25 25
6288- 25 2D 2C 2D 2D 35 2D 35
6290- 2D 2E 2E 2E 36 35 36 35
6298- 37 F5 36 37 3E 3E 3E 3E
62A0- 37 3F 37 27 37 1C 3F 3C
62A8- 27 27 27 27 3C 24 3C 2C
62B0- 3C 45 23 2C 24 25 25 25
62B8- 2D 2C 2D 35 2D 35 2D 2E
62C0- 2E 2E 36 35 36 36 3E 36
62C8- 37 37 37 3F 3E 3F 3F 27
62D0- 3F 3C 3C 3C 24 27 24 24
62D8- 2C 24 25 25 25 2D 2C 35
62E0- 2D 35 2D 2E 2E 2E 36 35
62E8- 36 3E 36 37 37 37 3F 3E
62F0- 3F 27 3F 3C 3C 3C 24 27
62F8- 24 2C 24 25 25 25 2D AC
6300- 2D 2E 35 35 35 35 2E 36
6308- 37 3E 3E 3E 3E 37 3F 3C
6310- 27 27 27 27 3C 24 25 2C
6318- 2C 2C 2C 2D 2E 35 35 35
6320- 35 2E 1E 3E 3E 3E 3E 37
6328- 1C 27 27 27 27 3C 0C 2C
6330- 2C 2C 2C 2E 35 35 35 35
6338- 3E 3E 3E 3E 27 27 27 27
6340- 2C 2C 2C 35 35 35 F5 37
6348- 37 1C 3C 3C 0C 0C 2E 2E
6350- 1E 1C 04 00 49 49 2D 2D
6358- 2D 2E 35 35 35 35 2E 36
6360- 36 36 37 3E 3E 3E 3E 37
6368- 3F 3F 3F 3C 27 27 27 27
6370- 3C 24 24 24 25 2C 2C 2C
6378- 2C 2D 2D 2D 2E 35 35 35
6380- 35 2E 36 36 37 3E 3E 3E
6388- 3E 37 3F 3F 3C 27 27 27
6390- 27 3C 24 24 25 2C 2C 2C
6398- 2C 2D 2D 2E 35 35 35 35
63A0- 2E 36 37 3E 3E 3E 3E 37
63A8- 3F 3C 27 27 27 27 3C 24
63B0- 25 2C 2C 2C 2C 2E 2C 2E
63B8- 35 35 35 35 2E 1E 3E 3E
63C0- 3E 3E 37 47 3B 3C 3C 3C
63C8- 24 67 2C 2C 2C 2D 35 35
63D0- 35 35 3E 3E 3E 3E 27 27
63D8- 27 27 2C 2C 2C 35 35 35
63E0- F5 37 37 47 23 27 67 0C
63E8- 36 0E 24 15 06 00 49 49
63F0- 2D 2D 2D 2E 35 35 35 35
63F8- 2E 36 36 36 37 3E 3E 3E
6400- 3E 37 3F 3F 3F 3C 27 27
6408- 27 27 3C 24 24 24 25 2C
6410- 2C 2C 2C AD 1B 2D 2D 2D
6418- 2D 55 3A 3F 3F 3F 3F 3F
6420- BF 2A 2D 2D 2D 2D 2D 2D
6428- AD 3A 3F 3F 3F 3F 3F 3F
6430- 3F B7 2D 2D 2D 2D 2D 2D
6438- 2D 2D D6 3F 3F 3F 3F 3F
6440- 3F 3F 56 2D 2D 2D 2D 2D
6448- 2D D6 3B 3F 3F 3F E7 00
6450- 49 49 2D 2D 2D 2E 35 35
6458- 35 35 2E 36 36 36 37 3E
6460- 3E 3E 3E 37 3F 3F 3F 3C
6468- 27 27 27 27 3C 24 24 24
6470- 25 2C 2C 2C 2C 6D 49 11
6478- 36 35 2E 36 35 2E 36 D6
6480- 27 3C 24 27 3C 24 27 3C
6488- 24 FC 13 36 35 2E 36 35
6490- 2E 36 35 2E 36 F5 1B 3C
6498- 24 27 3C 24 27 3C 24 27
64A0- 3C D6 36 35 2E 36 35 2E
64A8- 36 35 06 00 49 49 2D 2D
64B0- 2D 2E 35 35 35 35 2E 36
64B8- 36 36 37 3E 3E 3E 3E 37
64C0- 3F 3F 3F 3C 27 27 27 27
64C8- 3C 24 24 24 25 2C 2C 2C
64D0- 2C AD 1B 36 35 36 35 36
64D8- 3E 36 3E 36 4D 49 09 24
64E0- 27 24 27 24 2C 24 2C 24
64E8- DC 1B 36 36 36 36 36 36
64F0- 36 36 44 43 43 43 43 43
64F8- C3 1F 3F 3F 6F 49 49 69
6500- 2D 2D 25 00 49 49 2D 2D
6508- 2D 2E 35 35 35 35 2E 36
6510- 36 36 37 3E 3E 3E 3E 37
6518- 3F 3F 3F 3C 27 27 27 27
6520- 3C 24 24 24 25 2C 2C 2C
6528- 2C 2D 4D 69 FA 1B DF 73
6530- 69 09 8D DF FB 1B 9F 4D
6538- 69 09 4D F1 FB 1B DF FB
6540- 6A 09 4D 69 09 8D DF FB
6548- 1B DF FB 1B 4E 4D 69 09
6550- 4D 69 FA 1B DF FB 1B DF
6558- 73 69 09 4D 69 09 DE DF
6560- FB 1B 9F 4D 69 09 4D F1
6568- FB 1B DF 73 69 09 8D DF
6570- FB 1B 4E 4D 21 00 49 49
6578- 2D 2D 2D 2E 35 35 35 35
6580- 2E 36 36 36 37 3E 3E 3E
6588- 3E 37 3F 3F 3F 3C 27 27
6590- 27 27 3C 24 24 24 25 2C
6598- 2C 2C 2C 6D 69 89 3F 3F
65A0- 3F 3F 6E 09 4D 69 3A 3F
65A8- 3F 3F 3F 3F 77 69 09 4D
65B0- 69 11 3F 3F 3F 3F 3F 3F
65B8- 3F 17 4D 69 09 4D 69 09
65C0- 3E 3F 3F 3F 3F 3F 3F 3F
65C8- 37 4D 69 09 4D 69 09 3E
65D0- 3F 3F 3F 3F 3F 3F 3F 37
65D8- 4D 69 09 4D 69 09 1E 3F
65E0- 3F 3F 3F 3F 3F 3F 4E 69
65E8- 09 4D 69 3A 3F 3F 3F 3F
65F0- 3F 77 69 09 4D 31 3F 3F
65F8- 3F 3F 4E 09 04 00 49 29
6600- 2E 35 35 2E 3E 36 37 3F
6608- 3E 3C 27 27 3C 2C 24 25
6610- 35 0C 2E 35 35 36 37 3F
6618- 27 27 24 2D 35 35 37 27
6620- 27 25 00 09 AD 15 F6 1E
6628- 3F 38 20 0C 2D 2E 3E 3E
6630- 3C 2C 04 00 09 AD 15 F6
6638- 1E 3F 38 20 0C 04 00 20
```

智力棋的考驗



```
1  REM   :----------------------:
2  REM   :    MASTER MIND       :
3  REM   :  BY  MANVILLE CHAN   :
4  REM   :    COPYRIGHT  1985   :
5  REM   :  BY  COMPUTING AGE   :
6  REM   :----------------------:
10  FOR I = 0 TO  PEEK (78) / 5:A
     = RND (1): NEXT : IF  PEEK
    (24576) <  > 20 THEN  PRINT
     CHR$ (4);"BLOAD MASTER MIND
     SHAPES"
20  HGR : POKE  - 16302,0: SCALE=
    1: ROT= 0: POKE 232,0: POKE
    233,96
30  FOR A = 1 TO 80 STEP 16: FOR
    B = 0 TO 7: HCOLOR= 5: HPLOT
    1,A + B TO 120,A + B: NEXT :
     HCOLOR= 3: FOR C = 8 TO 16:
     HPLOT 1,A + C TO 120,A + C:
     NEXT C,A: HCOLOR= 2: FOR A =
    1 TO 40: HPLOT 1,A TO 60,A: NEXT
40  HCOLOR= 3: FOR A = 4 TO 56 STEP
    8: FOR B = 4 TO 36 STEP 8: DRAW
    10 AT A,B: NEXT B,A: FOR I =
    8 TO 52 STEP 8: FOR J = 8 TO
    32 STEP 8: DRAW 10 AT I,J: NEXT
    J,I
50  FOR I = 1 TO 6: DRAW I AT 30,
    I * 17 + 70: NEXT : DRAW 1 AT
    80,97: FOR I = 7 TO 9: DRAW
    I AT 80,I * 17 - 5: NEXT : HPLOT
    0,0 TO 120,0 TO 120,191 TO 0
    ,191 TO 0,0
60  FOR I = 770 TO 790: READ J: POKE
    I,J: NEXT : FOR A = 0 TO 31:
     READ I,J: POKE 768,I: POKE
    769,J * 32 - 1: CALL 770: FOR
    B = 1 TO 50: NEXT B,A
70  FOR I = 130 TO 279: HPLOT I,0
     TO I,191: NEXT : HCOLOR= 0:
     FOR I = 165 TO 250 STEP 28:
     FOR J = 25 TO 191 STEP 28: DRAW
    11 AT I,J: NEXT J,I: FOR I =
    25 TO 191 STEP 28: DRAW 18 AT
    132,I: DRAW 18 AT 145,I + 13
    : DRAW 18 AT 132,I + 13: DRAW
    18 AT 145,I: NEXT
80  FOR I = 1 TO 4:A(I) =  INT ( RND
    (1) * 6) + 1: NEXT : ON (A(1
    ) = A(2) AND A(2) = A(3)) OR
    (A(1) = A(2) AND A(1) = A(4)
    ) OR (A(1) = A(3) AND A(1) =
    A(4)) OR (A(2) = A(3) AND A(
    3) = A(4)) GOTO 80
90  FOR K = 6 TO 1 STEP  - 1: FOR
    J = 1 TO 4:C = 1
100 GOSUB 220: GET A$: IF A$ =  CHR$
    (8) THEN C = C - 1: IF C = 0
     THEN C = 6
110 IF A$ =  CHR$ (21) THEN C =
    C + 1: IF C = 7 THEN C = 1
120 IF A$ =  CHR$ (27) AND J > 1
     THEN  HCOLOR= 0: DRAW 12 AT
    J * 28 + 140,K * 28:J = J -
    2: NEXT
130 ON A$ <  >  CHR$ (13) GOTO 1
    00:C(J) = C: NEXT :X = 0:Y =
    0: FOR I = 1 TO 4:D(I) = I:E
    (I) = I: NEXT : FOR I = 1 TO
    4: IF A(I) = C(I) THEN D(I) =
    0:E(I) = 0:X = X + 1
140 NEXT : FOR I = 1 TO 4: FOR J
     = 1 TO 4: IF C(E(I)) = A(D(
    J)) AND C(E(I)) <  > 0 THEN
    E(I) = 0:D(J) = 0:Y = Y + 1
150 NEXT J,I:L = 0: IF Y > 0 THEN
     FOR I = 1 TO Y:L = L + 1:C =
    17: GOSUB 230: NEXT
160 IF X > 0 THEN  FOR I = 1 TO
    X:L = L + 1:C = 18: GOSUB 23
    0: NEXT
170 IF X <  > 4 THEN  NEXT
180 L = X: HCOLOR= 0: FOR I = 0 TO
    20: HPLOT 130,I TO 279,I: NEXT
    : FOR J = 1 TO 4:K = 0:C = A
    (J): GOSUB 220: NEXT : GET A
    $: HOME : TEXT : VTAB 5: HTAB
    10: PRINT "YOU LOSE!!": IF L
     = 4 THEN  VTAB 5: HTAB 10: PRINT
    "YOU WIN !!"
190 VTAB 10: PRINT "DO YOU WANT
    TO PLAY AGAIN ? ( Y OR N ) "
    ;
200 GET A$: IF A$ = "Y" THEN  RUN
210 ON A$ <  > "N" GOTO 200: HOME
    : END
220 A = J * 28 + 140:B = K * 28: HCOLOR= 0:
    DRAW 12 AT A,B: HCOLOR= 3: DRAW
    11 + C AT A,B: RETURN
230 HCOLOR= 3:J = (L = 2 OR L =
    4) + 1:M = (L > 2) + 1:A = J
     * 13 + 121:B = M * 13 + K *
    28 - 14: DRAW C + 2 AT A,B: RETURN :
    DATA 173,48,192,136,208,5,206 ,1
    ,3,240,9,202,208,245,174,0,3
    ,76,2,3,96
240 DATA 102,1,102,1,96,1,86,5,1
    02,1,102,1,115,1,128,6,136,1
    ,128,1,115,5,102,1,115,1,128
    ,1,115,6,172,1,172,1,128,1,1
    28,1,128,4,86,1,86,1,102,1,1
    15,1,128,4,128,2,115,6,154,2
    ,136,6,154,1,172,1,128,8
```

德國橋牌

— 賴耀華 —

德國橋牌是人數限制較少的一種橋牌遊戲，玩法較爲簡單，且十分普及。橋牌是多人遊戲，若要單人進行，自然要求助於電腦了。

這種橋牌的玩法多數人都懂得，在此不多作介紹。

這個程式主要是利用電腦模擬對手，即玩者與電腦鬥智。操作方法如下：

(一)起動程式；

(二)選擇對手數目，至少要有一個對手，多至八個，若選擇零，則返回BASIC；

(三)每次派牌前，電腦進行一次洗牌。派牌完畢，則螢幕下方顯示你手上的牌，S代表葵扇，H代表紅心，D代表鑽石，C代表梅花。

(四)左上角顯示皇牌，中間是可叫的最大墩數，右上角是局數。叫牌次序是第一局由你開始，之後每局輪轉一次，規則與德國橋牌完全一樣。

(五)跟着是打牌。打牌的方法是先按花後按點，然後確定輸入用RE－TURN。更正鍵是左箭咀。

(六)每局完成後，電腦進行計分。之後可選擇重新開始或繼續。

(七)每一次打牌之前可按ESCAP鍵，再輸入零，便可觀看曾經打出枱面的牌；輸入1代表自己手中的牌，其餘的是2－8，代表其餘對手的手中牌。其實這個設施是在設計除錯之時遺下來的工具。



這個橋牌程式的最重要部份是電腦叫牌和打牌。電腦叫牌是根據牌的強弱和剩餘墩數（即最大可叫墩數減去已叫的墩數）來進行。電腦將牌點化成分，而皇牌又較普通牌評分較高，再加上剩餘墩數與可叫的最大墩數比例乘以一個百分比而成。

電腦打牌部分是全程式中最長的。電腦决定出牌的大與小是根據取墩情況和枱面出牌的大小。過程與普通人思考大同小異，只不過少了蠱惑因素而已。

行100－160是洗牌；
行200－930是電腦思考打牌；
行1000－1055是遊戲開始；
行1060－1115是派牌；
行1120－1250是將牌分類；
行1300－1450是顯示玩者手牌；
行1500－1580是叫牌；
行1600－3090是打牌；
行3100－3350是比較和評分；
行5000－使亂數再排列；
行6000－6080是初始化；
行7000－7090是選擇對手人數；
行8000－8510是畫面顯示副常式；
行8600－8660是找尋最大的枱面牌；
行9000－9090是電腦叫牌；
行10000－10090是觀看各家手牌和全局出牌情況的副常式，這副常式是筆者在觀察程式運作時用的工具，用來測驗電腦的出牌是否合理。

本文程式只有一個，GERMAN BRIDGE，大家可照列表輸入或在同期出版的程式磁碟內找到相同的檔案。■

```
1   REM  ******************
2   REM  * GERMAN BRIDGE *
3   REM  *  LAI YIU WAH  *
4   REM  *   (C)1985  BY  *
5   REM  * COMPUTING AGE *
6   REM  ******************
10  GOTO 1000
100  REM  SHUFFLING
101  HOME
102  VTAB 11: HTAB 15: FLASH : PRINT
     "SHUFFLING"
104  FOR I = 1 TO 52:CARD(I) = 0:
     NEXT
110  FOR I = 1 TO 52
120 RN =  INT ( RND (R) * 52) + 1
130  IF CARD(RN) THEN 120
135 CARD(RN) = I
140  NEXT
150  HOME : NORMAL
160  RETURN
200  REM  FIRST PLAY
205  IF RD = 1 AND BID(CO) *  RND
     (R) < ROUND / 2 THEN  GOSUB
     400: RETURN
210  IF BB(CO) <  BID(CO) THEN  GOSUB
     240
220  IF BB(CO) >  = BID(CO) THEN
     GOSUB 400
230  RETURN
240  IF TR > 0 AND PS(CO,TR,1) >
     0 THEN 290
250 MT = 1: FOR I = 2 TO 4: IF PS
     (CO,I,1) > PS(CO,MT,1) THEN
     MT = I
260  NEXT :SO = MT:VA = PS(CO,MT,
     1): RETURN
290  IF PS(CO,TR,1) > 9 THEN SO =
     TR:VA = PS(CO,TR,1): RETURN
300  FOR I = ROUND TO 1 STEP  - 1
     : IF PS(CO,TR,I) > 0 THEN MT
      = I:I = 1
310  NEXT :SO = TR:VA = PS(CO,TR,
     MT)
390  RETURN
400 VA = 14: FOR I = 1 TO 4: FOR
     J = 1 TO ROUND: IF (PS(CO,I,
     J) < VA) AND PS(CO,I,J) > 0 THEN
     SO = I:VA = PS(CO,I,J)
```

```
410  NEXT : NEXT
420  RETURN
500  REM  FOLLOW
505  GOSUB 8600
510  IF BB(CO) < BID(CO) THEN  GOSUB
     600
520  IF BB(CO) >  = BID(CO) THEN
      GOSUB 800
530  RETURN
600  REM
610  IF PS(CO,MS(DD),1) = 0 THEN
     700
615  IF MS(MM) <  > MS(DD) THEN 6
     50
620  FOR I = ROUND TO 1 STEP  - 1
     : IF PS(CO,MS(DD),I) > MV(MM
     ) THEN VA = PS(CO,MS(DD),I):
     I = 1
630  NEXT : IF VA <  > 0 THEN 670

640  GOTO 650
650  FOR I = ROUND TO 1 STEP  - 1
     : IF PS(CO,MS(DD),I) > 0 THEN
     VA = PS(CO,MS(DD),I):I = 1
660  NEXT
670  SO = MS(DD)
680  RETURN
700  IF (MS(DD) <  > TT) AND PS(C
     O,TT,1) <  > 0 AND MS(MM) =
     TT THEN 750
705  VA = 14
710  FOR I = 1 TO 4: FOR J = 1 TO
     ROUND: IF (PS(CO,I,J) < VA) AND
     PS(CO,I,J) > 0 THEN VA = PS(
     CO,I,J):SO = I
720  NEXT : NEXT
730  RETURN
750  FOR I = ROUND TO 1 STEP  - 1
     : IF PS(CO,TT,I) > MV(MM) THEN
     VA = PS(CO,TT,I):SO = TT:I =
     1
760  NEXT : IF VA <  > 0 THEN 770

762  FOR I = 1 TO 4: IF MS(DD) =
     I OR TT = I THEN 766
763  FOR J = ROUND TO 1 STEP  - 1
     : IF PS(CO,I,J) <  > 0 THEN
     VA = PS(CO,I,J):SO = I
764  NEXT
766  NEXT : IF VA <  > 0 THEN 770

767  FOR I = ROUND TO 1 STEP  - 1
     : IF PS(CO,TT,I) <  > 0 THEN
     VA = PS(CO,TT,I):I = 1
768  NEXT :SO = TT
770  RETURN
800  IF PS(CO,MS(DD),1) = 0 THEN
     900
810  IF MS(MM) = MS(DD) THEN 850
820  VA = PS(CO,MS(DD),1):SO = MS(
     DD): RETURN
850  FOR I = 1 TO ROUND: IF PS(CO
     ,MS(DD),I) < MS(MM) THEN VA =
     PS(CO,MS(DD),I):I = ROUND
860  NEXT :SO = MS(DD): IF VA = 0
      THEN VA = PS(CO,MS(DD),1)
870  RETURN
900  FOR I = 1 TO 4: IF MS(DD) =
     I OR TT = I THEN 920
910  IF (PS(CO,I,1) > VA) THEN VA
      = PS(CO,I,1):SO = I
920  NEXT : IF VA = 0 THEN VA = P
     S(CO,TT,1):SO = TT
930  RETURN
1000  REM  START
1010  GOTO 7000
1015  GOSUB 6000
1020  HOME
1030  ROUND = 0:INC = 1:DEALER = 0

1035  RR = 0
1040  ROUND = ROUND + INC: IF ROUN
      D > MAXRD THEN INC =  - 1: GOTO
      1040
1050  IF ROUND = 0 THEN INC = 1: GOTO
      1040
1051  RR = RR + 1
1052  DEALER = DEALER + 1: IF DEAL
      ER > PNUMBER THEN DEALER = 1

1055  POKE 35,24: GOSUB 100
1060  REM  DISTRIBUTION
1070  COUNT = 1
1080  FOR I = 1 TO ROUND
1090  FOR J = 1 TO PNUMBER
1100  PCARD(J,I) = CARD(COUNT)
1110  COUNT = COUNT + 1: NEXT : NEXT

1115  FOR I = 1 TO PNUMBER: FOR J
      = 1 TO 4: FOR K = 1 TO ROUN
      D:PS(I,J,K) = 0: NEXT : NEXT
      : NEXT
1120  REM  SORTING
1130  FOR I = 1 TO PNUMBER
1140  FOR J = 1 TO 4
1145  PM = 1
1150  FOR K = 1 TO ROUND
1160  IF CS(PCARD(I,K)) <  > J THEN
      1250
1180  PF = 1
1190  IF CV(PCARD(I,K)) < PS(I,J,
      PF) THEN PF = PF + 1: IF PF <
      = PM THEN 1190
1200  IF PF > PM THEN PM = PF: GOTO
      1240
1210  FOR L = PM TO PF STEP  - 1
1220  PS(I,J,L + 1) = PS(I,J,L)
1230  NEXT
1235  PM = PM + 1
1240  PS(I,J,PF) = CV(PCARD(I,K))
1245  PCARD(I,K) = 0
1250  NEXT K,J,I
1300  REM  SHOW CARD
1310  PP = 1: GOSUB 8300
1400  REM  CHOOSE TRUMP
1405  VTAB 3: GOSUB 8500
1410  RN =  INT ( RND (R) * 52 + 1
      )
1415  VTAB 3: HTAB 1
1420  TRUMP = CS(RN): IF CV(RN) =
      13 THEN TRUMP = 0: PRINT "NO
      ";: GOTO 1440
1430  PRINT A$(TRUMP);
1440  PRINT ".TRUMP"
1450  VTAB 3: HTAB 33: PRINT "ROU
      ND="RR
1500  REM  BIDDING
1510  HOME
1520  COUNT = DEALER:CC = 1:BID =
      0
1525  FOR I = 1 TO PNUMBER:BID(I)
      = 0:BB(I) = 0: NEXT
1526  VTAB 3: HTAB 13: PRINT "MAX
      .BID="ROUND
```

```
1530  IF COUNT = 1 THEN  INPUT "Y
   OU BID ";BID(1): IF BID(1) >
   ROUND OR BID(1) < 0 THEN 153
   0
1535  IF COUNT = 1 AND CC = PNUMB
   ER AND BID(1) + BID = ROUND THEN
   1530
1540  IF COUNT <  > 1 THEN  GOSUB
   9000: PRINT "PLAYER "COUNT"
   BID "BID(COUNT)
1545 BID = BID + BID(COUNT)
1550 COUNT = COUNT + 1: IF COUNT >
   PNUMBER THEN COUNT = 1
1560 CC = CC + 1: IF CC <  = PNUM
   BER THEN 1530
1570 PP = 1: GOSUB 8200
1580  VTAB 3: HTAB 13: PRINT "BID
   ING="BID"/"ROUND
1600  REM  PLAYING
1610 DD = DEALER:RD = 1
1620 CO = DD:CC = 1
1625  HOME :X = 1:Y = 5
1630 VA = 0:SO = 0: IF CO <  > 1 THEN
   2000
1640  HTAB X: VTAB Y + CO: PRINT
   "YOU     ";
1645  IF RD = ROUND THEN  GOSUB 4
   00:PA = VA: GOSUB 8000: PRINT
   A$(SO) CHR$ (PA): GOTO 3000
1650  GET Z$: IF Z$ =  CHR$ (27) THEN
   GOSUB 10000: GOTO 1640
1660  IF (Z$ = "S") + (Z$ = "H") +
   (Z$ = "D") + (Z$ = "C") = 0 THEN
   1640
1670  FOR I = 1 TO 4: IF A$(I) =
   Z$ THEN SO = I
1680  NEXT : IF DD <  > 1 AND PS(
   CO,MS(DD),1) > 0 AND SO <  >
   MS(DD) THEN 1640
1685  PRINT Z$;
1690  GET Z$:PA =  ASC (Z$): GOSUB
   8100: IF Z$ =  CHR$ (8) THEN
   1640
1692  IF PA < 1 OR PA > 13 THEN 1
   690
1695 VA = PA
1700  PRINT Z$;
1710  GET Z$: IF Z$ =  CHR$ (13) THEN
   1740
1720  IF Z$ =  CHR$ (8) THEN 1640
1730  GOTO 1710
1740  FOR I = 1 TO ROUND:TEST = 0
   : IF (PS(1,SO,I) = VA) THEN
   TEST = 1:I = ROUND
1750  NEXT : IF TEST = 0 THEN 164
   0
1760  GOTO 3000
2000  REM  COMPUTER PLAY
2005 VA = 0:SO = 0
2010  IF CO = DD THEN  GOSUB 200
2020  IF CO <  > DD THEN  GOSUB 5
   00
2030  HTAB X: VTAB Y + CO: PRINT
   "PLAYER "CO" ";:PA = VA: GOSUB
   8000: PRINT A$(SO) CHR$ (PA)
3000 MS(CO) = SO:MV(CO) = VA
3005 CARD((SO - 1) * 13 + VA) = 0
3010  FOR I = 1 TO ROUND: IF (PS(
   CO,SO,I) = VA) THEN PF = I:I
   = ROUND
3020  NEXT :PS(CO,SO,PF) = 0: IF
   PF = ROUND THEN 3060
3030  FOR I = PF TO ROUND: IF I =
   MAXRD THEN 3050
3040 PS(CO,SO,I) = PS(CO,SO,I + 1
   )
3050  NEXT :PS(CO,SO,ROUND) = 0
3060 CC = CC + 1: IF CC > PNUMBER
   THEN 3100
3070 CO = CO + 1: IF CO > PNUMBER
   THEN CO = 1
3090  GOTO 1630
3100  REM  MATCHING
3110  GOSUB 8600
3190 BB(MM) = BB(MM) + 1
3195 DD = MM
3200  HTAB X + 9: VTAB Y + MM: INVERSE
   :PA = MV(MM): GOSUB 8000: PRINT
   A$(MS(MM)) CHR$ (PA): NORMAL

3205 PP = 1: GOSUB 8200: GOSUB 83
   00
3210  GOSUB 8400: REM  WAIT
3230 RD = RD + 1: IF RD > ROUND THEN
   3250
3240  GOTO 1620
3250  FOR I = 1 TO PNUMBER
3270 SC(I) = SC(I) + (BB(I) = BID
   (I)) * (BB(I) ^ 2 + 10) - (B
   B(I) <  > BID(I)) * ( ABS (B
   B(I) - BID(I))) ^ 2
3280  NEXT
3290  HOME : PRINT : PRINT "TOTAL
   SCORE:": PRINT : FOR I =.1 TO
   PNUMBER: IF I > 1 THEN  PRINT
   "PLAYER "I;: GOTO 3310
3300  PRINT "YOU.....";
3310  PRINT ".."SC(I);: HTAB 20: IF
   BB(I) = BID(I) THEN  PRINT "
   MAKE "BB(I)
3314  IF BB(I) > BID(I) THEN  PRINT
   "UP   "BB(I) - BID(I)
3316  IF BB(I) < BID(I) THEN  PRINT
   "DOWN "BID(I) - BB(I)
3320  NEXT
3330  GOSUB 8400: REM  WAIT
3335  HOME : VTAB 10
3340  PRINT : INVERSE : PRINT "[E
   SCAPE] TO NEW [OTHER] TO CON
   TINUE ";: NORMAL : GET Z$: IF
   Z$ =  CHR$ (27) THEN 1000
3350  GOTO 1040
5000 R =  INT ( RND (R) * 60000):
   RETURN : REM  RANDOMIZE
6000  REM  INITIALIZE
6005  DIM CARD(52),CS(52),CV(52),
   SC(PNUMBER),BID(PNUMBER),PCA
   RD(PNUMBER,MAXRD),PS(PNUMBER
   ,4,MAXRD),A$(4),BB(PN),MS(PN
   ),MV(PN)
6020 SO = 1
6030  FOR I = 0 TO 3
6040  FOR J = 1 TO 13
6050 CS(I * 13 + J) = SO:CV(I * 1
   3 + J) = J
6060  NEXT :SO = SO + 1: IF SO =
   5 THEN SO = 1
6070  NEXT
6075 A$(1) = "S":A$(2) = "H":A$(3
   ) = "D":A$(4) = "C"
6080  GOSUB 8400: RETURN
```

```
7000  REM  HEADING
7010  TEXT : HOME : CLEAR
7020  PRINT "***  GERMAN BRIDGE
 by Lai Yiu Wah    ***"
7030  POKE 34,3
7040  VTAB 7: PRINT "YOU ARE ONE
 OF THE PLAYERS IN THE GAME."
7045  PRINT
7050  PRINT "COMPUTER ACTS AS THE
 OTHER PLAYERS."
7055  PRINT
7060  INPUT "NO. OF RIVALS(1-8):"
 ;PNUMBER:PNUMBER = PNUMBER +
 1: IF PN = 1 THEN  TEXT : HOME
 : END
7070  IF PNUMBER < 2 OR PNUMBER >
 9 THEN 7060
7080 MAXRD =  INT (52 / PNUMBER)
7090  GOTO 1015
8000 PA = (PA < 9) * 49 + (PA = 9
 ) * 75 + (PA = 13) * 52 + (P
 A = 10) * 64 + (PA = 11) * 7
 0 + (PA = 12) * 63 + PA: RETURN
 : REM  CHANGING
8100 PA = PA - (PA > 49 AND PA <
 58) * 49 - (PA = 84) * 75
 (PA = 65) * 52 - (PA = 74) *
 64 - (PA = 81) * 70 - (PA =
 75) * 63: RETURN : REM  CHA
 RACTOR TO VALUE
8200  REM  SHOW BID
8205  POKE 34,18: POKE 35,24
8210  VTAB 19: GOSUB 8500
8220  VTAB 19: HTAB 1: IF PP = 1 THEN
 PRINT "YOU";: GOTO 8235
8230  PRINT "PLAYER."PP;
8235  HTAB 10: PRINT "GET."BB(PP)
 ;
8240  HTAB 34: PRINT "BID."BID(PP
 )
8245  POKE 34,3: POKE 35,18
8250  RETURN
8300  REM  SHOW CARD
8310  POKE 34,19: POKE 35,24: HOME

8320  FOR I = 1 TO 4: VTAB 19 + I
 : HTAB 1: PRINT A$(I)":";
8330  FOR J = 1 TO ROUND
8340  IF PS(PP,I,J) > 0 THEN PA =
 PS(PP,I,J): GOSUB 8000: PRINT
 CHR$ (PA);
8350  NEXT J,I
8360  POKE 34,3: POKE 35,18
8370  RETURN
8400  REM  WAIT
8410  VTAB 17: HTAB 1: INVERSE : PRINT
 "PRESS ANY KEY"
8420  IF  PEEK ( - 16384) < 128 THEN
 R =  INT ( RND (1) * 60000):
 GOTO 8420
8430  POKE  - 16368,0: NORMAL
8440  RETURN
8500  HTAB 1: PRINT "...........
 ..........................
 "
8510  RETURN
8600  REM  FIND THE BIGGEST
8610 MM = DD:TT = TRUMP: IF TT =
 0 THEN TT = MS(DD)
8614 CR = 0
8615  FOR I = 1 TO PNUMBER: IF MS
 (I) <  > TT THEN CR = CR + 1
8616  NEXT : IF CR = PNUMBER THEN
 TT = MS(DD)
8620  FOR I = 1 TO PNUMBER
8630  IF (MS(I) = TT) AND (MS(MM)
 <  > TT) THEN MM = I: GOTO
 8650
8640  IF MS(I) = MS(MM) THEN  IF
 MV(I) > MV(MM) THEN MM = I
8650  NEXT
8660  RETURN
9000  REM  COMPUTER BID
9010  FOR I = 1 TO 4
9015 VA = 0
9020  FOR J = 1 TO ROUND
9030 VP = PS(COUNT,I,J)
9035  IF VP = 0 THEN 9070
9040 FA = (ROUND - BID) / ROUND *
 CC / PNUMBER * 13 + VA / 13 +
 (VA > 13) * ROUND * .3 + (RO
 UND - BID) / ROUND * (MAXRD -
 ROUND) / MAXRD * 13
9050  IF TRUMP = I THEN VA = VA +
 (VP > 9) * VP + (VP < 10) *
 VP * .5 + FA
9055  IF TRUMP = 0 THEN VA = VA +
 VP * .1 + FA
9060  IF TRUMP <  > I AND TRUMP <
 > 0 THEN VA = VA + (VP > 11
 ) * (VP - (MAXRD - ROUND) /
 MAXRD * 7.5)
9070  NEXT J:BID(COUNT) = BID(COU
 NT) +  INT (VA / 13)
9075 ADD = ADD + (VA / 13)
9080  NEXT I
9085 BID(COUNT) = (BID(COUNT) > 0
 AND BID(COUNT) <  = ROUND) *
 BID(COUNT) + (BID(COUNT) > R
 OUND) * ROUND
9086  IF CC <  > PNUMBER THEN 909
 0
9087 BID(COUNT) = (BID(COUNT) + B
 ID = ROUND) * ((ADD -  INT (
 ADD) >  = .6) - (ADD -  INT
 (ADD) < .6)) + BID(COUNT): IF
 BID(COUNT) < 0 THEN BID(COUN
 T) = 1
9088  IF BID(COUNT) > ROUND THEN
 BID(COUNT) = BID(COUNT) - 2
9090  RETURN
10000  POKE 34,18: POKE 35,24: VTAB
 19: HTAB 1: PRINT "........
 ...........................
 ..."
10005  VTAB 19: HTAB 1: PRINT "(
 1-"PNUMBER") OR ( 0 ) OVERAL
 L ";: INPUT PP: IF PP < 0 OR
 PP > PNUMBER THEN 10000
10010 PP =  INT (PP): IF PP > 0 THEN
 GOSUB 8200: GOSUB 8300: GOTO
 10090
10015  VTAB 19: GOSUB 8500
10020  FOR I = 1 TO 4: VTAB 19 +
 I: HTAB 1: PRINT A$(I)":";
10030  FOR J = 1 TO 13: IF CARD((
 I - 1) * 13 + J) <  > 0 THEN
 PRINT " ";: GOTO 10060
10040 PA = J: GOSUB 8000: PRINT  CHR$
 (PA);
10060  NEXT : NEXT
10090  POKE 34,3: POKE 35,18: RETURN
```

CRAZY PROTECTION

ALAN LEE

磁碟的保護（PROTECTION）可說是千變萬化，而保護的「程度」亦參差不齊。

在現時最常見及最勁的PROTECTION可說是SPIR—AL TRACK PROTECTION了。

我爲了使大家對PROTECTION方面能有一定的了解及興趣，所以我特別寫了一個非常簡單而且是利用NORMA—L DOS 3·3來作PROTECTION的PROGRAM。

這個程式所PROTECT的方法，在表面看着實非常簡單，但如果各位參詳眞正的原理，然後再以其他方面再MO—DIFY的話，所得出來的結果，可能會令你大出意料之外。

這PROGRAM所PROTECT的方法，我在這處不作詳細介紹，因爲只要大家打入程式，然後再LIST來看，便會了解得一清二楚了。而這PROTECTION的特色是他可以被任何一個COPY PROGRAM所「COPY」到，但是除了COPYA及之類的COPY PROGRAM；但當一BOOT 起那隻BACKUP的磁碟時，只見磁碟機的臂被人「點穴」一般，動也不動而磁碟機卻「長轉」。

大家看完了以上的介紹，可能會以爲我「誇大」，但只要大家打入以下程式，並執行之，便可知道誰是誰非。不過，我所說的是以不改動任何PRAM爲準。而我曾試過以下數個COPY勁PROGRAM在不改動任何PRAM情況下，不能COPY到由這個CRAZY PROTECTION PR—OGRAM所製造出來的成品。這些PROGRAM是：LO—CKSMITH V5·0，E·D·D III COPY II PLUS VERSION 5·0，NADOL——NIBBLES AWAY III。

閒話休說，我現在爲大家介紹怎樣使用這個所謂勁的CRAZY PROTECTION之PROGRAM。

（一）首先BLOAD你想PROTECT的FILE GA—ME，但有一點要注意的是所PROTECT之遊戲的長度，不能長過9100。換句話說可以PROTECT之PROGR—AM的地址是限於800——90FF止。

（二）BLOAD CRAZY PROTECTION A$9100。

（三）將所要PROTECT之程式的起始地址（STAR—TION ADDRESS）放在$91FE及$91FF處。而輸入之方法是有HI BYTE及LO BYTE之分別。舉一例來說，如果遊戲之STARTING是1D00，那麼我們只需要鍵入91FE：00 1D便可。

（四）按9200G 來運行此程式。

（五）隔一會兒，螢幕會顯示你要使用那一部磁碟機作PROTECTION之用。如果是DRIVE1，只要鍵入「一」。

（六）放入空碟在你所需要用的磁碟機中，並按RETU—RN鍵。

（七）當FORMAT完空碟後，此程式會自動將要保護的程式WRITE入磁碟內。

（八）當以上動作做完後，此程式會問你：「TEXT GRAPHIC」，意思是你想當BOOT起這BACKUP時，電腦所要設定的狀態。如果選擇TEXT的話，只要按「T」便可以。

（九）如果使用TEXT時，我提供了一個不能超過12個字符的HEADING供大家使用，大家只要打入你所希望在BOOT起碟時，在SCREEN所顯示之字句，然後按RETURN便可。

（十）在GRAPHIC方面，大家只要按「G」便可。而在GRAPHIC狀態，我提供了HGR 1及HGR 2之選擇，如果你希望設定HGR 2的話，只要按「2」便能夠了。

（十一）當選擇完以上的程序時，亦卽是你的BACKU—P製造完成的當兒。

由於以上程式寫得急一點，所以可能會有少許漏洞及不方便。而我所發現的是忘記設定BACKUP DISK之RE—SET POINTER，所以我在此補充：

在9200G ——RUN這程式時先在以下修改：

```
B6B0:A9 00 8D F2 03 A9 C6 8D F3 03 20 6F FB 60
B63F:20 B0 08
```

最後如果大家對此程式有何意見，請來信多多指教。

```
**********************
* CRAZY PROTECTION  *
*     VERSION 1     *
*   A$9100,L$500    *
*        BY         *
*     ALAN LEE      *
*  (C) COPYRIGHT    *
*  COMPUTING AGE    *
**********************
9100- A2 00 BD 96 08 9D 01 07
9108- E8 E0 0F D0 F5 A9 00 85
9110- F0 A9 07 85 F1 85 F3 A9
9118- 01 85 F2 A2 00 A0 0D B1
9120- F0 91 F2 88 C0 FF D0 F7
9128- A9 F0 20 A8 FC EE 7E 08
9130- E8 E0 0C D0 E8 60 A0 A0
9138- A0 A0 A0 A0 A0 A0 A0 A0
9140- A0 A0 A0 A0 A0 A0 A0 A0
9148- A0 A0 A0 A0 A0 A0 A0 A0
9150- A2 00 A9 00 9D 00 20 E8
9158- D0 F8 EE 66 08 AD 66 08
9160- C9 60 D0 EE AD 50 C0 AD
9168- 57 C0 AD 52 C0 AD 54 C0
9170- AD 55 C0 60 A9 B6 8D 82
9178- 91 A9 10 8D 85 91 A2 00
9180- BD 00 B6 9D 00 10 E8 D0
```

CRAZY PROTECTION

```
9188- F7 EE 82 91 EE 85 91 AD
9190- 82 91 C9 C0 D0 EA 60 A2
9198- 00 BD 00 91 9D 60 B6 E8
91A0- E0 50 D0 F5 60 A9 50 8D
91A8- 9A 91 20 97 91 20 B8 93
91B0- C9 B2 F0 09 C9 B1 D0 F5
91B8- A9 60 8D 80 B6 60 A9 00
91C0- 8D 9A 91 20 97 91 20 E3
91C8- 95 A9 BA 85 33 20 F7 92
91D0- AD 0D 02 D0 F1 A2 00 BD
91D8- 00 02 9D 97 B6 E8 E0 0D
91E0- D0 F5 60 A9 58 8D 43 B6
91E8- A9 FC 8D 44 B6 A9 60 8D
91F0- 46 B6 A9 08 8D 47 B6 A9
91F8- B6 20 C8 93 60 00 00 00
9200- 20 AF 93 A2 00 A9 04 A0
9208- B4 20 00 94 A2 11 A9 05
9210- A0 37 20 00 94 A2 1C A9
9218- 05 A0 B5 20 00 94 A2 2B
9220- A0 38 A9 06 20 00 94 A2
9228- 34 A0 B5 A9 06 20 00 94
9230- A0 20 A9 FF 20 A8 FC 88
9238- D0 F8 20 18 94 A2 11 A0
9240- 08 A9 04 20 00 94 C8 E8
9248- 20 08 94 A2 7C A9 07 A0
9250- 07 20 00 94 20 F6 94 C9
9258- B1 90 F9 C9 B3 B0 F5 8D
9260- FE 94 AD 10 C0 20 D0 93
9268- 20 F6 94 C9 8D D0 F9 20
9270- E5 93 20 F0 93 20 00 93
9278- 20 7C 93 20 E5 93 A9 BE
9280- 8D F1 93 20 F0 93 20 76
9288- 93 20 40 93 20 7C 93 20
9290- E5 93 A9 4E 8D F1 93 A9
9298- 55 8D FB 93 20 F0 93 20
92A0- F6 94 C9 D4 F0 0B C9 C7
92A8- D0 F5 20 A5 91 A9 02 D0
92B0- 03 20 BE 91 AD FE 91 8D
92B8- 42 B7 AD FF 91 20 D9 92
92C0- A9 00 8D ED B7 8D F0 B7
92C8- 8D EC B7 A9 10 8D F1 B7
92D0- 8D E3 B7 20 5A 93 4C A2
92D8- 93 8D 43 B7 A9 38 8D 42
92E0- B9 A9 0A 8D 15 B7 A9 01
92E8- 8D 1A B7 A9 92 8D E7 B7
92F0- 20 E3 91 20 74 91 60 AD
92F8- 10 C0 20 AF 94 60 60 00
9300- AD FE 94 8D EA B7 A9 04
9308- 8D F4 B7 A9 18 8D 42 B9
9310- A9 4C 8D CE BD A9 BC 8D
9318- CF BD A9 BD 8D D0 BD A9
9320- AA 8D 91 B9 8D AE BC A9
9328- 20 8D B3 BC A9 00 8D EB
9330- B7 8D F6 B7 AD E5 B7 AC
9338- E4 B7 20 B5 B7 60 00 00
9340- A9 89 8D E0 B7 8D E3 B7
9348- A9 01 8D EC B7 A9 00 8D
9350- F0 B7 A9 08 8D ED B7 8D
9358- F1 B7 20 34 93 AC ED B7
9360- C8 C0 10 D0 05 A0 00 EE
9368- EC B7 8C ED B7 EE F1 B7
9370- CE E3 B7 D0 E5 60 A9 02
9378- 8D F4 B7 60 AD F5 B7 C9
9380- 10 F0 0D C9 20 F0 09 C9
9388- 40 F0 05 C9 08 F0 01 60
9390- 20 E5 93 A2 CB A9 06 A0
9398- B0 20 00 94 AD 10 C0 20
93A0- F6 94 A9 B3 8D F1 93 A9
93A8- 98 8D FB 93 4C A9 94 A9
93B0- 00 8D F5 B7 20 4C 94 60
93B8- 20 E5 93 A2 6C A9 06 A0
93C0- B0 20 00 94 20 F6 94 60
93C8- 8D FE B6 20 D4 94 60 A9
93D0- A2 43 A9 06 A0 AA 20 00
93D8- 94 C8 A2 98 20 08 94 C8
93E0- E8 20 08 94 60 A2 27 A9
93E8- A0 9D A7 06 CA D0 F8 60
93F0- A2 B3 A9 06 A0 B0 20 00
93F8- 94 C8 A2 98 20 08 94 60
9400- 8C 0E 94 8D 0F 94 A0 00
9408- BD 00 95 F0 0A 99 B0 06
9410- 20 E4 94 C8 E8 D0 F1 60
9418- A9 18 85 FE A2 17 8A 20
9420- C1 FB A5 28 85 06 A5 29
9428- 85 07 CA 8A 20 C1 FB A0
9430- 27 B1 28 91 06 88 10 F9
9438- E0 00 F0 02 10 E4 A0 27
9440- A9 A0 91 28 88 10 FB C6
9448- FE D0 D1 60 A9 28 85 FE
9450- A2 00 A0 00 8A 20 C1 FB
9458- C8 B1 28 88 91 28 C8 C0
9460- 27 D0 F5 A9 A0 91 28 E8
9468- E0 18 D0 E6 C6 FE D0 E0
9470- 60 A0 14 84 FE A2 00 A0
9478- 12 8A 20 C1 FB B1 28 C8
9480- 91 28 88 F0 03 88 10 F5
9488- A9 A0 91 28 A0 15 B1 28
9490- 88 91 28 C8 C0 27 F0 03
9498- C8 10 F3 A9 A0 91 28 E8
94A0- E0 18 D0 D3 C6 FE D0 CD
94A8- 60 20 71 94 4C 00 92 A9
94B0- 00 85 25 A2 00 A9 8D 20
94B8- ED FD E8 E0 0F D0 F6 20
94C0- 6A FD A2 00 BD 00 02 C9
94C8- 8D F0 03 E8 D0 F6 A9 A0
94D0- 9D 00 02 60 A9 6C 8D 4A
94D8- B6 A9 FD 8D 4B B6 A9 08
94E0- 8D 4C B6 60 84 F0 A0 20
94E8- A9 70 20 A8 FC AD 30 C0
94F0- 88 D0 FD A4 F0 60 AD 00
94F8- C0 C9 80 90 F9 60 B1 A9
9500- C3 D2 C1 DA D9 A0 D0 D2
9508- CF D4 C5 C3 D4 C9 CF CE
9510- 00 D7 D2 C9 D4 D4 C5 CE
9518- A0 C2 D9 00 41 4C 41 4E
9520- 20 4C 45 45 20 43 2E 20
9528- 4B 2E 00 BC C3 BE A0 B1
9530- B9 B8 B5 00 C3 CF CD D0
9538- D5 D4 C9 CE C7 A0 C1 C7
9540- C5 AE 00 C9 CE D3 C5 D2
9548- D4 AE AE BD BE 00 DB D4
9550- DD C5 D8 D4 00 DB C7 DD
9558- D2 C1 D0 C8 C9 C3 00 C5
9560- CE D4 C5 D2 A0 CE C1 CD
9568- C5 BD BE 00 C8 C7 D2 DB
9570- B1 DD A0 CF D2 A0 DB B2
9578- DD A0 BF 00 D5 D3 C5 A0
9580- D7 C8 C9 C3 C8 A0 C4 D2
9588- C9 D6 C5 A0 DB B1 DD A0
9590- CF D2 A0 DB B2 DD BF 00
9598- C2 C1 C3 CB D5 D0 A0 C4
95A0- C9 D3 CB 00 A0 A6 A0 C8
95A8- C9 D4 A0 DB D2 C5 D4 DD
95B0- A0 60 00 C6 CF D2 CD C1
95B8- D4 D4 C9 CE C7 00 D7 D2
95C0- C9 D4 D4 C9 CE C7 00 C4
95C8- CF D3 00 D0 CC C5 C1 D3
95D0- C5 A0 C3 C8 C1 C3 CB A0
95D8- D9 CF D5 D2 A0 C4 D2 C9
95E0- D6 C5 00 20 E5 93 A2 5F
95E8- A9 06 A0 B0 20 00 94 A2
95F0- 00 A9 A0 9D 00 02 E8 D0
95F8- F8 A9 00 8D 0D 02 60 00
9600- 0D
```

將CPS 4、4C拆成FILE

蘋果新丁

自COPY][PLUS 5·0 出版以來，提供了用戶多種新功能如改良SECTOR EDITOR和AUTO COPY等，對於磁碟編輯已是不可缺少的工具。執筆之時，COPY][PLUS 已出版至版本5·4 了。至於已擁有新版本的用戶，舊版本如4·4C是否已沒有存在價值？那也未必。因爲自版本5·0起，COPY][PLUS 的出版公司已放棄了舊有的COPY PROTECTION，而本人手上的原版4·4C 仍保留着保護，這對喜愛拆碟的用戶極具挑戰。以下便是本人將COPY][PLUS 4·4C 的UTILITY和BIT COPY 合併後存入一個FILE 的方法。

準備工具

1.插在SLOT 0 的16K RAM CARD
2.一隻已FORMAT 的空碟
3.原版COPY][PLUS 4·4C

解拆過程

此段集中應付解拆過程中出現的難題，實對解拆其他磁片有很大的幫助。對此沒有興趣的讀者可直接跳至「解拆撮要」一段。

首先BOOT 起4·4C，當出現主MENU後，嘗試按RESET 。嗶一聲後仍操作自如，顯示RESET VECTOR已指向程式開始的ADDRESS，使用戶不能RESET至MO－NITOR。但BOOTDISK 這功能可幫助用戶跳至MON－ITOR。

選擇BOOT DISK，按<CR>
DRIVE 啓動後按兩次RESET

此時APPLESOFT PROMPT應顯示於左下方。

CALL－151<CR>

由$800開始LIST。當LIST至$851 時發現JMP $11AD。再LIST $11AD時發現一連串ZERO PAGE 和PAGE THREE 的初值。似乎是程式開始的ADDRESS。由於按RESET 時程式仍運行自如，所以檢視RESET VEC－TOR 3F2·3F3<CR>

發覺是03 E0。即RESET VECTOR 指向$ E003。故估計程式是由$851 開始。再檢視程式，知道程式存放在$851至$4B2F。綜合以上資料，現試圖將UTILITY 存入FILE。先將第8頁放在安全ADDRESS以BOOT DOS

5800<800·8FFM<CR>
放入空碟BOOT DOS
6 CTRL P<CR>

將第8頁移回原址並SAVE 上碟

800<5800·58FFM<CR>
BSAVE UTILITY，A$851，L$42DF

關上電源，再開啓並BRUN UTILITY。現在難題出現了。一、CURSOR BAR並不是放在CATALOG，而是BO－OT DISK。二、SLOT NO 是F而不是6。 前者是由於跳往MONITOR 時是用BOOT DISK 這功能，所以CUR－SOR 是停放在BOOT DISK。解決不難，只要修改RE－SETVECTOR，使RESET 後能跳至MONITOR，再放回CURSOR原有位置，接着寫回RESET VECTOR 便成。至於後者是由於$11C7 是LDA $2FF。而$2FF是存放於SLOT NO×16。即$60。但由於剛才已關上電源，所以$2FF 存着的不是$60而是$FF。解決也不難，要先將$60、存放在$2FF 便成。

BLOAD UTILITY
11B7：EA EA EA<CR>
11BF：EA EA EA<CR>
11C4：EA EA EA<CR>
3F2：69 FF 5A<CR>
11ADG
將CURSOR 放回CATALOG
RESET
11B7：8D F2 03<CR>
11BF：8D F3 03<CR>
11C4：8D F4 03<CR>
84C：A9 60 8D FF 02
3D0G
BSAVE UTILITY，A $84C，L$42E4

關上電源，再開啓並BOOT DOS。接着試一試BRUN UTILITY。一切運行自如。

解拆BIT COPY 就容易得多了。BOOT 起BIT COPY 後按Q以BOOT DISK。放入空碟，再按<CR>。因爲BIT COPY 是放在由位址$900起，所以無需保留PAGE 8。而$900就是程式開始，同樣先將$60放在$2FF 再開始執行程式8FB：A9 60 8D FF 02

存放入碟

BSAVE BIT COPY，A $8FB，L$2969

解拆完成。

合併兩檔案

現在討論如何將兩個FILE 合併了。由於COPY][PLUS UTILITY 將利用$4E00 以上作VIEW FILE 及其他功能用的BUFFER，所以如果將BIT COPY 放在48K記憶中便會被破壞。唯一解決的辦法是先將BIT COPY 放上RAM CARD，然後執行UTILITY。當選擇BIT COPY時，便將RAM CARD 上的BIT COPY 放回原址執行即可。

將CPS 4、4C拆成FILE

經過檢視程式，發覺$4B40至4B80 是安全ADDRE－SS可存放這個MOVE ROUTINE。

至於原有執行BIT COPY時DISK ACCESS的ADD－RESS是$1AE1。由於現在執行BIT COPY時已無需再放入磁片，所以本人將程式略作修改，以刪去此命令。

最後將此FILE存放入碟便大功告成。在此我建議大家用PRONTO－DOS或其他近似DOS 3·3 因爲UTILI－TY是應用DOS的RWTS作READ OR WRITE的。

解拆撮要

解拆 UTILITY

1. BOOT起UTILITY
2. 選擇BOOT DISK，按<CR>
3. 打開機門
4. 按<CR>
5. 按RESET兩次
6. CALL－151<CR>
7. 5800<800·8FFM<CR>
8. 放入空碟
9. 6 CTRL P<CR>
10. CALL－151<CR>
11. 800<5800·58FFM<CR>
12. 11B7：EA EA EA <CR>
    11BF：EA EA EA <CR>
    11C4：EA EA EA <CR>
    3F2：69 FF 5A <CR>
    11ADG<CR>
13. 將CURSOR BAR放回CATALOG
14. RESET
15. 11B7：8D F2 03<CR>
    11BF：8D F3 03<CR>
    11C4：8D F4 03<CR>
    84C：A9 60 8D FF 02<CR>
16. 3D0G<CR>
17. BSAVE UTILITY，A$84C，L$42E4

```
4B40- 2C 81 C0 2C 81 C0 A0 00
4B48- 84 42 84 3C 84 3E A9 D0
4B50- 85 43 A9 4C 85 3D A9 76
4B58- 85 3F 20 2C FE 2C 82 C0
4B60- 60 2C 80 C0 A2 00 BD 00
4B68- D0 9D 00 09 E8 D0 F7 EE
4B70- 68 4B EE 6B 4B AD 68 4B
4B78- C9 FA D0 EA 2C 82 C0 4C
4B80- 00 09
```

表(一)

解拆BIT COPY

1. BOOT起BIT COPY
2. 按Q
3. 放入空碟，按<CR>
4. CALL－151<CR>
5. 8FB：A9 60 8D FF 02 <CR>
6. BSAVE BIT COPY，A$8FB，L $2969

合併

1. BOOT DOS
2. BLOAD UTILITY，A $84C
3. BLOAD BIT COPY，A $4BFB
4. CALL－151<CR>
5. 849：20 40 4B<CR>
6. 1AE1：4C 61 4B<CR>
7. 1AA2：4C C6 1A<CR>
8. 鍵入表一
9. BSAVE COPY][PLUS 4·4C，A$849，L$6D1B

警告

本人已將COPY][PLUS 4·4C 解拆了半年多了，而一切亦運行自如。但爲了安全起見，大家應保留原版本，以免招致損失。■

刋登 電腦時代 廣告

請電：3－312007

機械碼輸入校對工具

本文程式CHECKSUM TABLE主要是幫助大家於鍵入本刊刊出的機械語言程式後供偵錯參考。鑑於有些機械語言並未錄程式落磁碟上，故此，大家在將程式鍵入後，最好宜先BSAVE 起，然後再利用這個CHECKSUM TABLE 程式印出剛鍵入的機械語言列表之CHECKSUM ，跟列表右邊之CHECKSUM 值核對一次，假如兩者一樣，則表示鍵入無錯，假如有出入，則表示該行有某些數值鍵入錯誤。

CHECKSUM TABLE 程式的用法十分方便，首先是BRUN CHECKSUM TABLE，再將要CHECK 的機械語言程式載入，然後鍵入CALL－151 進入監督程式內，然後將要CHECK的機械碼開始地址及結束地址鍵入，再按入〔CTRL－Y〕 RETURN，就以本文程式為一個例子。

＊ 300・3CE，〔CTRL－Y〕〔RETURN〕，這時螢幕便會將CHECKSUM 在每八個位元組後顯示出，若果要配合印字機使用的話，可以在執行程式前先啓動印字機（PR＃1）然後再依照以上步驟去做即可。

最後一提的是有些機種沒有CTRL－Y 向量的，所以要採用以下方法：

＊ 300・3CE N 321 G〔RETURN〕

希望各位能好好利用此程式供日後鍵入機械語言後作偵錯參考之用。

鍵入本程式方法，大家可參閱第十七期「如何輸入機械語言」一文。或在同期出版的程式磁碟內找出相同名字的FILE。■

```
*******************
* CHECKSUM TABLE *
*  A$300, L$CF   *
*       BY       *
*   COMPUHOUSE   *
*******************
0300- 20 58 FF BA CA BD 00 01  $B2E1
0308- 18 69 1F 8D F9 03 85 62  $286C
0310- E8 BD 00 01 69 00 8D FA  $D2ED
0318- 03 85 63 A9 4C 8D F8 03  $68E2
0320- 60 20 8E FD A9 0A 85 0A  $2066
0328- A0 00 84 31 20 A7 FF A9  $5284
0330- FF 85 31 A5 3C 85 0B A5  $0223
0338- 3D 85 0C 20 A7 FF A0 55  $D448
0340- A9 10 91 62 A9 FB C8 91  $6CDF
0348- 62 A0 00 F0 45 A5 3C 29  $E502
0350- 07 D0 42 38 A9 1F E5 24  $E2B6
0358- AA 20 4A F9 A9 A4 20 ED  $3E81
0360- FD A5 0B A6 0C 20 41 F9  $459D
0368- C6 0A D0 26 20 8E FD A9  $2DD5
0370- 0A 85 0A AD 00 C0 10 FB  $C644
0378- 8D 10 C0 C9 83 F0 48 C9  $DD18
0380- A0 F0 BB C9 9B D0 0B A9  $F4A9
0388- EA A0 55 91 62 C8 91 62  $3194
0390- A0 00 20 92 FD A9 A0 20  $68CA
0398- ED FD B1 3C 48 20 DA FD  $F970

03A0- 68 6A 45 0B 2A 45 0C 85  $5DC3
03A8- 0B 45 0C 6A 85 0C 20 BA  $1DC9
03B0- FC 90 9A A9 1F E5 24 AA  $5D93
03B8- 20 4A F9 A9 A4 20 ED FD  $95A4
03C0- A5 0B A6 0C 20 41 F9 20  $B271
03C8- 8E FD 8D 10 C0 60 FF     $E387
```

第二十期HELLO BREAKER修改

第廿期的HELLO BREAKER 是我渴望很久的程式，方便而實用。但它有很輕微的缺點。當按下＜ＥＳＣ＞鍵後，檔案的次序反白字母〝A〞，是由螢光幕的第一行開始，剛好重叠在〝DISK VOLUME〞的一行上，觀感上不是十全十美。我提議把程式的第180行至第210 行作以下的變更。不知閣下意見如何？

編者註：多謝讀者溫煜讚君來函將第20期刊出的HELLO BREAKER 作出修改，使之更「十全十美」。

```
180 PRINT"CATALOG";:HTAB 16:PRINT
"<ESC> = STOP CATALOG
190 VTAB 1:YN=0
200 D$ = CHR$ (4):PRINT D$;"CATALOG"
202 B=PEEK (37) -2
203 IF B<21 THEN M =4:GOTO 210
204 B=21:M =0
210 T=0:CH=1: FOR CV=M TO 23:GOSUB
100: IF C=193 OR C=194 OR C=201
OR C=212 THEN POKE P+5,T+1:T=
T+1:S=CV
```

預防勝於治療

磁碟壽命性能測試器

沈德成

如果你有一張滿載重要資料的磁碟，突然出現I／O ERROR，令這磁碟內的程式不能運作，眞是心痛死人，雖然可以利用BAG OF TRACK 或其他救碟程式來救碟，但由於這些程式都不可以READ到原有這些ERROR SECT－OR（S）內的資料，通常這些救碟程式會將這些ERROR SECTOR（S）格式化（FORMAT），但裏面的資料會因格式化而流失，這樣，一是你的程式不能運行，一是有BUG（S），所以始終都是預防勝於治療。但坊間中，根本沒有一個程式有預防I／O ERROR 的作用，因此我便寫一個有預防功能的程式──DISK CILLIN了。

預防原理

當一片新磁片開始使用後，其性能便開始受到塵埃、磁場、自然衰退、意外損傷等影响，經過一段時間後便逐漸退化了。但是那一張磁片退化至什麽程度呢？我們是無從知道的，直至出現I／O ERROR的那一刻。

當RWTS 讀取一個SECTOR 時，可能因瞬間的磁碟機運轉不穩或電訊干擾而無法讀取，RWTS 對於這種情形便會重複嘗試讀48次，若仍無法讀取，則令磁頭歸零以保位置正確，再重新至該磁區嘗試48次，若總共96次都不能讀取，RWTS才放棄而宣告I／O ERROR。

利用DOS這種特性，便可由DOS的嘗試次數來判斷磁片狀況，對於超過16次（$10）的磁片，其性能已有問題。

運作與及使用方法

如果想知道DOS的讀取次數，就必須在RWTS 上做手脚，首先我們要在$BE6D 做一個指標去到CHECK ，即是將原本的BDC9：DE $578 改爲BDC9：JSR，而在CHECK 這段程式中，會去計出DOS 的讀取次數，然後將結果放在TEMP 的位置中，最後做回DEC$578 的工作，再返回RWTS，因爲要使用RWTS ，爲了方便使用，所以我寫了一段機械碼程式，而它是在CHECK 程式之下，由於這兩個副程式非常有用，大家可於日後任意搬移位置。本程式的用法方面，在ASSEMBLY 列表中，已有詳細介紹，所以不在此詳說，主程式是用BASIC 寫成，當運行BA－SIC主程式，它會將副程式LOAD 進，所以你要將主副兩個程式，放在同一隻磁碟，否則會出現FILE NOT FOU－ND 的錯誤訊息，首先它會問你要測試的磁碟是35（$22軌），還是40（$27）軌，你只要適當地按"1"或"2"，然後放入要測試的磁碟放進DRIVE 1 ，並且按下<RE－TURN>，接着便開始測試，它會將各條磁軌的狀態寫出，分別是：

"NO ERROR"──沒有問題

"ILLED"──即是這條軌有病，你必需盡快及早將它治療，免至令這條軌將來DEAD了。

"DEAD"──即是這條軌已死了，這時只有用INIT（BAG OF TRACKS），將壞了的SEC－TOR IS從新FORMAT，但裏面的資料，將會流失。

最後程式會將ILLED和DEAD了的磁軌總數寫出，並且會問你需否測試其它磁碟。

首先鍵入BASIC 部份，然後用SAVE DISK CILL－IN 命令存入磁碟，接着便要鍵入機械碼部份，然後用BSAVE CHECK & RWTS UTILITY，A$300，L $53存入磁碟。（當然，如果懂滙編語言的讀者，可直接用LISA 2·5或MERLIN打入滙編碼。）

各位亦可在同期出版的程式磁碟上找到相同的檔案名字。■

```
1   REM  *******************
2   REM  *    DISK CILLIN  *
3   REM  *       BY        *
4   REM  * SHUM TAK SHING  *
5   REM  *******************
10  TEXT : HOME
20  VTAB 12: PRINT  TAB( 9)"PROGR
    AM IS LOADING": PRINT  TAB(
    14)"PLEASE WAIT A MINUTE"
30  PRINT  CHR$ (4);"BLOADCHECK &
    RWTS UTILITY.OBJ"
40  HOME :A$ = " DISK CILLIN":B =
    LEN (A$): FOR I = 1 TO B +
    12: VTAB 1: IF I > B THEN  HTAB
    I - B
41  FOR WA = 1 TO 50: NEXT : PRINT
    RIGHT$ (A$,I): NEXT
42  POKE 34,1: VTAB 24: PRINT  TAB(
    17)"BY": PRINT : PRINT  TAB(
    13)"JOHNNY SHUM": FOR I = 1 TO
    18: PRINT : NEXT
43  FOR I = 1 TO 2000: NEXT
44  POKE 34,0
50  HOME
80  CALL 768: REM INIT CHECK
85  POKE 47083,0: REM DISK VOLUME
     0,THEN CAN TEST ALL OF DOS
    DISK
90  HOME
```

磁碟壽命性能測試器

```
100  PRINT "T";: PRINT  TAB( 13);
     : INVERSE : PRINT "DISK CILL
     IN": NORMAL :A$ = "012345678
     9ABCDEF": PRINT "R": PRINT "
     K" TAB( 17)"1" TAB( 33)"2": PRINT
     A$;A$; LEFT$ (A$,8)
110  POKE 34,5
115  VTAB 7: PRINT "CHECK WHAT TR
     ACK DISK ?": PRINT " 1. 35($
     22) TRACKS DISK": PRINT " 2.
     40($27) TRACKS DISK": VTAB
     22: PRINT "PRESS >1< FOR 35"
     : PRINT "       >2< FOR 40";
120  GET TR$: IF TR$ <  > "1" AND
     TR$ <  > "2" THEN  PRINT : VTAB
     22: PRINT "YOU MUST ENTER '1
     ' OR '2'": PRINT "PLEASE PRE
     SS ANY KEY TO CONTIUNE";: GET
     A$: HOME : VTAB 10: PRINT "-
     >";: GOTO 120
130  IF TR$ = "1" THEN TR = 34
140  IF TR$ = "2" THEN TR = 39
150  HOME : VTAB 7: PRINT "PUT YO
     UR TEST DISK IN DRIVE 1": PRINT
     : PRINT "PRESS ANY KEY TO ST
     ART THE TEST";: GET A$
160  HOME
165  POKE 2,0
166  IL = 0:DE = 0
170  FOR T = 0 TO TR
175  POKE 0,T
180  FOR S = 0 TO 15
190  IF  PEEK (2) <  > 0 THEN  NEXT
     : GOTO 220
200  POKE 1,S
210  CALL 825
215  NEXT
220  IF  PEEK (2) = 0 AND  PEEK (
     822) > 16 THEN  PRINT "I": PRINT
     "L": PRINT "L": PRINT "E": PRINT
     "D": POKE 822,0:IL = IL + 1:
     GOTO 250
230  IF  PEEK (2) <  > 0 THEN  PRINT
     "D": PRINT "E": PRINT "A": PRINT
     "D": POKE 2,0:DE = DE + 1: GOTO
     250
240  IF  PEEK (2) = 0 AND  PEEK (
     822) <  = 16 THEN  PRINT "N"
     : PRINT "O": PRINT : PRINT "
     E": PRINT "R": PRINT "R": PRINT
     "O": PRINT "R"
250  POKE 32,T + 1: VTAB 6: NEXT
260  POKE 32,0: VTAB 17
270  PRINT "TOTAL ILLED:";IL: PRINT
     "TOTAL DEAD :";DE: PRINT "TO
     TAL ERROR:";IL + DE
280  VTAB 22: PRINT "ANOTHER TEST
     ?(Y/N) ";: GET A$: IF A$ <
     > "Y" AND A$ <  > "N" THEN
     GOTO 280
290  IF A$ = "Y" THEN  HOME : GOTO
     120
300  TEXT : HOME : PRINT "GOOD BY
     E !": PRINT : END
```

```
0800                *CHECK
0800                *
0300                          ORG $300
0300                ;
0300                ;CHANGE DOS DIRECTION TO CHECK
0300                ;
0300 A9 20                    LDA #$20
0302 8D C9 BD                 STA $BDC9
0305 A9 24                    LDA #CHECK
0307 8D CA BD                 STA $BDCA
030A A9 03                    LDA /CHECK
030C 8D CB BD                 STA $BDCB
030F                ;
030F                ;INIT READING
030F                ;
030F A9 00                    LDA #$00
0311 8D F0 B7                 STA $B7F0
0314 A9 20                    LDA #$20
0316 8D F1 B7                 STA $B7F1
0319 A9 01                    LDA #$01
031B 8D F4 B7                 STA $B7F4           ;READ
031E                ;
031E                ;CLEAR TEMP
031E                ;
031E A9 00                    LDA #$00
0320 8D 36 03                 STA TEMP
0323 60                       RTS
0324                ;
0324                ;CHECK
0324                ;
0324 A9 30          CHECK     LDA #$30
0326 38                       SEC
0327 ED 78 05                 SBC $578
032A CD 36 03                 CMP TEMP
032D 90 03                    BLT RETURN1
032F 8D 36 03                 STA TEMP
0332 CE 78 05       RETURN1   DEC $578
0335 60                       RTS
0336                TEMP      DFS 1
0337                *
0337                *
0337                *RWTS UTILITY
03D9                RWTS      EQU $3D9
03E3                GETIOB    EQU $3E3
0002                USERERR   EPZ $02
0337                ;
0337                ;BEGIN OF ROUTIUE
0337                ;
0337 A5 00          START     LDA $00
0339 8D EC B7                 STA $B7EC
033C A5 01                    LDA $01
033E 8D ED B7                 STA $B7ED
0341 20 E3 03                 JSR GETIOB
0344 20 D9 03                 JSR RWTS
0347 90 05                    BCC EXIT
0349 AD F5 B7                 LDA $B7F5
034C 85 02                    STA USERERR
034E A9 00          EXIT      LDA #$00
0350 85 48                    STA $48
0352 60                       RTS
0353                          END

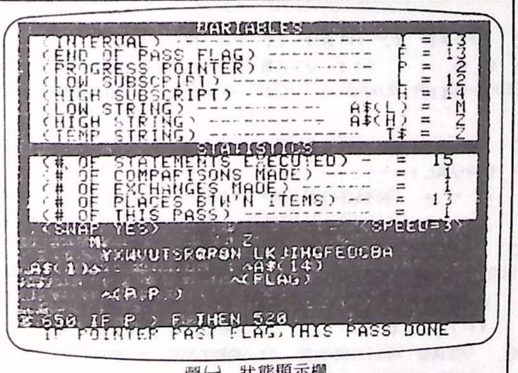
***** END OF ASSEMBLY
```

排序原理透視

學習程式編寫的過程中，懂得如何把資料快速及有效率地排序（SORT）是比較重要的一環。本文的一個程式TR－ANSORT ，可以使各位對排序的各個步驟及所需的技巧加深理解。

排序一般分爲兩種，數字及文字。雖然兩種都涉及了比較（COMPARE）和交換（SWAP），但是在處理的方法上便有所分別了。這是因爲電腦把數字以一組代碼看待，而文字則單獨每個字符化爲其ASC Ⅱ 值才能處理。故此作文字上的排序較爲複雜，電腦會首先把兩個字（WORD）的第一個字母（LETTER） 作比較，若兩者相同再比較第二個字，餘此類推。

介紹了排序的背景後，現在則解釋如何使用本程式。假如擁有程式磁碟或已鍵入列表的讀者可以用RUN TRANS－ORT 來執行本程式。沒多久便可看見主目錄了，你可以用左右箭咀及<RETURN>鍵來選擇目錄的TRANSPAR－ENT SORT，NON TRANSPARENT SORT，SORT LISTING 或END PROGRAM。首先可選擇TRANSPAR－ENT SORT 。



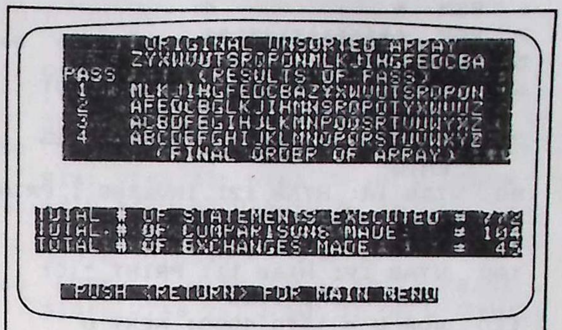
圖㈠ 狀態顯示欄

TRANSPARENT SORT

這項功能使你能看到排序過程中每個步驟，各個變數的值及一些資料等。之後你會看到第二個目錄，使你能選擇要排序的一個字串，你大可逐一試試，觀看對排序有什麽影响。選擇同樣是以左、右箭咀及<RETURN> 進行，無論選擇了那個字串，最後是會由A至Z 順序排列好的。

跟着便會看到一個狀態顯示表（見圖一），在此畫面可以看到了排序的執行。你可以自由選擇執行速度，以1爲最快及9爲最慢，預置是3，改變速度按該數字鍵便可。按<SPACE> 鍵則使其一步步執行，而按<M>則是停止執行而回到主目錄。

執行排序之時，那個字串是放在畫面中央，而一些字母的次序會隨着而不斷變動，下方則是現時執行中的BASIC命令及一些解釋。就這樣地觀看整個排序過程便會更加了解排序是如何工作的。



圖㈡ 排序過程各部份狀態及統計資料

基本上，整個排序分爲四部份，第一部份是每個字母與其相隔13的字母來比較，如果次序不符就使其交換次序。第二部份是相距 6 個字母，第三部份相距 3 個字母，而最後的是 1 個字母。在上方可以清清楚楚看到各個變數的狀態及工作情形。而最後會顯示了各部份的狀態及一些統計資料（圖二）。

NON－TRANSPARENT SORT

這項功能顯示了眞正排序時的時間，由於不需作任何狀態顯示，故可看出遠較TRANSPARENT SORT 爲快。

SORT LISTING

這項功能能讓你觀看排序的副程式（參看列表二）。這種方法由D・F・SHELL 提出，而由M・METZNER 改良，故名爲SHELL／METZNER SORT ，是一種甚有效率的排序方法，此處應用作26個字母的排序。

本文只有一個程式，讀者可照列表一鍵入後以SAVE TRANSORT 把程式存在磁碟中。■

```
510 I = 26
520 I =  INT (I / 2)
530  IF I = 0 THEN  RETURN
540 F = 26 - I
550 P = 1
560 L = P
570 H = L + I
580  IF A$(L) <  = A$(H)  THEN 640
590 T$ = A$(L)
600 A$(L) = A$(H)
610 A$(H) = T$
620 L = L - I
630  IF L > 0 THEN 570
640 P = P + 1
650  IF P > F THEN 520
660  GOTO 560
```

列表㈡ 排序時會利用到的副程式

```
1  REM ***************
2  REM *             *
3  REM *  TRANSORT   *
4  REM *     BY      *
5  REM * COMPUHOUSE *
6  REM *             *
7  REM ***************
50  HOME
60  VTAB 8: HTAB 8: PRINT "COMPUT
    ING AGE PUBLISHER"
70  VTAB 10: HTAB 16: PRINT "PRES
    ENTS"
80  VTAB 14: HTAB 12: INVERSE : PRINT
    "TRANSPARENT SORT"
90  NORMAL
100  VTAB 19: HTAB 11: PRINT "(C)
     COPYRIGHT 1985"
120  FOR U = 1 TO 4000: NEXT U
1100  REM  === MAIN MENU ===
1120 TI$ = ">>> MAIN MENU <<<":C =
     4:C$(1) = "TRANSPARENT SORT"
     :C$(2) = "NON-TRANSPARENT SO
     RT":C$(3) = "SORT LISTING":C
     $(4) = "END PROGRAM"
1130  GOSUB 9200
1140  ON K GOSUB 5000,7000,3000,1
      500
1150  GOTO 1100
1500  REM  === END PROGRAM ===
1510  HOME : HTAB 15: VTAB 15: PRINT
      "> E N D <"
1520  FOR X = 1 TO 1000: NEXT : VTAB
      20: END
3000  REM  === LIST SORT ===
3010  HOME : HTAB 3: INVERSE : PRINT
      ">> SHELL/METZNER SORTING RO
      UTINE <<";: NORMAL : PRINT
      : PRINT
3020  FOR X = 1 TO 16: ON X GOSUB
      6210,6220,6230,6240,6250,626
      0,6270,6280,6290,6300,6310,6
      320,6330,6340,6350,6360: PRINT
      : NEXT X
3030  VTAB 20: HTAB 3: FLASH : PRINT
      "  (PUSH ANY KEY TO CONTINUE
      ) ";: GET I$: NORMAL : RETURN

5000  REM  === SUPERVISE TRANS SO
      RT ===
5010  CLEAR : GOSUB 8100
5020  GOSUB 8200
5030  GOSUB 5200
5040  GOSUB 5400
5050  GOSUB 5100
5060  GOSUB 5500
5070  GOSUB 6900
5080  GOTO 1100
5100  REM  === PAUSE TO START ===

5110  FLASH : VTAB 23: HTAB 9: PRINT
      "PUSH ANY KEY TO START";
5120  GET I$: INVERSE :I$ =  STR$
      (SC): VTAB 16: HTAB 30: PRINT
      "<SPEED=" + I$ + ">";: NORMAL
      : RETURN
5200  REM  === DRAW SCREEN ===
5210  HOME : INVERSE
5220 VL$ = " " +  CHR$ (10) +  CHR$
     (8):VM$ = VL$ + VL$ + VL$ +
     VL$ + VL$ + VL$ + VL$ + VL$ +
     VL$ + VL$:VM$ = VM$ + VM$
5230  VTAB 1: PRINT "
        VARIABLES
     "
5231  VTAB 10: PRINT "
        STATISTICS
     "
5232  VTAB 17: HTAB 2: PRINT S1$;

5233  VTAB 18: HTAB 2: PRINT S1$;

5234  VTAB 19: HTAB 2: PRINT S1$;

5235  VTAB 20: HTAB 2: PRINT S1$;

5236  VTAB 21: HTAB 2: PRINT S1$;

5237  VTAB 22: HTAB 2: PRINT S1$;

5238  VTAB 23: HTAB 2: PRINT S1$;

5240  VTAB 2: HTAB 1: PRINT VM$ +
     VL$ + " ";: VTAB 2: HTAB 40:
      PRINT VM$ + VL$ + " ";: NORMAL

5250  VTAB 2: HTAB 3: PRINT "(INT
     ERVAL) ------------------  I
     = " +  RIGHT$ (" " +  STR$ (
     I),2);
5260  VTAB 3: HTAB 3: PRINT "(END
      OF PASS FLAG) ---------  F
     = " +  RIGHT$ ("  " +  STR$
     (F),2);
5270  VTAB 4: HTAB 3: PRINT "(PRO
     GRESS POINTER) ---------  P
     = " +  RIGHT$ ("  " +  STR$
     (P),2);
5280  VTAB 5: HTAB 3: PRINT "(LOW
      SUBSCRIPT) ------------  L
     = " +  RIGHT$ ("  " +  STR$
     (L),2);
5290  VTAB 6: HTAB 3: PRINT "(HIG
     H SUBSCRIPT) -----------  H
     = " +  RIGHT$ ("  " +  STR$
     (H),2);
5300  VTAB 7: HTAB 3: PRINT "(LOW
      STRING) ------------  A$(L)
     =  " + A$(L);
5310  VTAB 8: HTAB 3: PRINT "(HIG
     H STRING) ----------  A$(H)
     =  " + A$(H);
5320  VTAB 9: HTAB 3: PRINT "(TEM
     P STRING) -------------  T$
     =  " + T$;
```

```
5330  VTAB 11: HTAB 3: PRINT "(#
   OF STATEMENTS EXECUTED) -  =
    " + RIGHT$ ("   " +  STR$
   (RS),3);
5340  VTAB 12: HTAB 3: PRINT "(#
   OF COMPARISONS MADE) ----  =
    " + RIGHT$ ("   " +  STR$
   (RC),3);
5350  VTAB 13: HTAB 3: PRINT "(#
   OF EXCHANGES MADE) ------  =
    " + RIGHT$ ("   " +  STR$
   (RE),3);
5360  VTAB 14: HTAB 3: PRINT "(#
   OF PLACES BTW'N ITEMS) --  =
    " + RIGHT$ ("   " +  STR$
   (I),3);
5370  VTAB 15: HTAB 3: PRINT "(#
   OF THIS PASS) -----------  =
    " + RIGHT$ ("   " +  STR$
   (D),3);
5380  INVERSE : VTAB 16: HTAB 2: PRINT
   S1$;
5390  RETURN
5400  REM  === DISPLAY ALPHABET =
   ==
5410  INVERSE : VTAB 18: HTAB 7: GOSUB
   9000: NORMAL : RETURN
5500  REM  === SORT CONTROL ===
5510 I = 26:V = 1: GOSUB 5800
5520 I =  INT (I / 2):V = 2: GOSUB
   5800
5530 V = 3: GOSUB 5800: IF I = 0 THEN
   RETURN
5540 F = 26 - I:V = 4: GOSUB 5800
5550 P = 1:V = 5: GOSUB 5800
5560 L = P:V = 6: GOSUB 5800
5570 H = L + I:V = 7: GOSUB 5800
5580 V = 8: IF A$(L) <  = A$(H) THEN
   SW$ = "<SWAP NO> ": GOSUB 58
   00: GOTO 5640
5585 V = 8: IF A$(L) > A$(H) THEN
   SW$ = "<SWAP YES>": GOSUB 58
   00
5590 T$ = A$(L):V = 9: GOSUB 5800
5600 A$(L) = A$(H):V = 10: GOSUB
   5800
5610 A$(H) = T$:V = 11: GOSUB 580
   0
5620 L = L - I:V = 12: GOSUB 5800
5630 V = 13: GOSUB 5800: IF L > 0
   THEN 5570
5640 P = P + 1:V = 14: GOSUB 5800
5650 V = 15: GOSUB 5800: IF P > F
   THEN 5520
5660 V = 16: GOSUB 5800: GOTO 556
   0
5800 REM  === CHART HOUSEKEEPING
   ===
5810  NORMAL : REM  (STMT COUNT)
5820 RS = RS + 1: VTAB 11: HTAB T
   H: PRINT  RIGHT$ ("   " +  STR$
   (RS),3);
5830  REM  (UPDATE / STATS / VARS
   )
5840  ON V GOSUB 6010,6020,6030,6
   040,6050,6060,6070,6080,6090
   ,6100,6110,6120,6130,6140,61
   50,6160
5850  INVERSE : VTAB 23: HTAB 2: PRINT
   S1$: VTAB 23: HTAB 3: ON V GOSUB
   6210,6220,6230,6240,6250,626
   0,6270,6280,6290,6300,6310,6
   320,6330,6340,6350,6360
5860  NORMAL : VTAB 24: HTAB 2: PRINT
   S1$;: VTAB 24: HTAB 3: ON V GOSUB
   6410,6420,6430,6440,6450,646
   0,6470,6480,6490,6500,6510,6
   520,6530,6540,6550,6560: INVERSE
5900  REM  === STEP/SPEED CONTROL
   ===
5910 IF FL = 1 THEN 5970
5920  FOR X = 1 TO S:SC =  PEEK (
   - 16384): POKE  - 16368,0: IF
   SC > 127 THEN I$ =  CHR$ (SC
   - 128): IF I$ = " " THEN FL
   = 1: GOTO 5970
5930  IF SC > 127 AND I$ =  > "1"
   AND I$ <  = "9" THEN S =  VAL
   (I$) ^ 2: GOTO 5980
5935  IF SC > 127 AND I$ = "M" THEN
   CLEAR : GOTO 1100
5940  NEXT X: GOTO 5980
5970  VTAB 16: HTAB 30: INVERSE :
   PRINT "  <STEP>";: VTAB 16
   : HTAB 34: GET I$:: IF I$ =
   > "1" AND I$ <  = "9" THEN
   S =  VAL (I$) ^ 2:FL = 0
5980  IF FL = 0 THEN  INVERSE : IF
   I$ =  > "1" AND I$ <  = "9" THEN
   VTAB 16: HTAB 30: PRINT "<S
   PEED=" + I$ + ">";
5990  RETURN
5995  RETURN
6000  REM  === UPDATE VARIABLES =
   ==
6010  NORMAL : VTAB 2: HTAB HT: PRINT
   RIGHT$ ("  " +  STR$ (I),2)
   ;: RETURN
6015  STOP
6020  VTAB 2: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (I),2);:D = D +
   1: VTAB 14: HTAB TH: PRINT  RIGHT$
   ("  " +  STR$ (I),3);: GOSUB 670
   0: IF D < 5 THEN  VTAB 15: HTAB
   TH: PRINT  RIGHT$ ("   " +  STR$
   (D),3);: RETURN
6025  RETURN
6030  RETURN
```

```
6040  VTAB 3: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (F),2);: INVERSE
   : VTAB 20: HTAB 2: PRINT S1$
   ;: VTAB 20: HTAB (6 + F): PRINT
   "^(FLAG)";: NORMAL : RETURN

6050  VTAB 4: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (P),2);: INVERSE
   : VTAB 21: HTAB 2: PRINT S1$
   ;: VTAB 21: HTAB (6 + P): PRINT
   "^(P.P.)";: NORMAL : RETURN
6060  VTAB 5: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (L),2);: RETURN

6070  VTAB 6: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (H),2);: RETURN

6080  GOSUB 6800:RC = RC + 1: VTAB
   12: HTAB TH: PRINT  RIGHT$ (
   "  " +  STR$ (RC),3);: RETURN

6090  VTAB 9: HTAB HT: PRINT " " +
   T$;: RETURN
6100  VTAB 7: HTAB HT: PRINT " " +
   A$(L);: INVERSE : VTAB 17: HTAB
   6 + L: PRINT A$(L);: NORMAL
   : RETURN
6110  VTAB 8: HTAB HT: PRINT " " +
   A$(H);: INVERSE : VTAB 17: HTAB
   6 + H: PRINT A$(H);: NORMAL
   :RE = RE + 1:: VTAB 13: HTAB
   TH: PRINT  RIGHT$ ("  " +  STR$
   (RE),3);: RETURN
6120  VTAB 5: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (L),2);: RETURN

6130  RETURN
6140  VTAB 4: HTAB HT: PRINT  RIGHT$
   ("  " +  STR$ (P),2);: INVERSE
   : VTAB 21: HTAB 2: PRINT S1$
   ;: VTAB 21: HTAB (6 + P): PRINT
   "^(P.P.)";: RETURN
6150  RETURN
6160  RETURN
6200  REM  === LINE STATEMENTS ==
   =
6210  PRINT "510 I = 26";: RETURN

6220  PRINT "520 I = INT (I / 2)"
   ;: RETURN
6230  PRINT "530 IF I = 0 THEN RE
   TURN";: RETURN
6240  PRINT "540 F = 26 - I";: RETURN

6250  PRINT "550 P = 1";: RETURN

6260  PRINT "560 L = P";: RETURN

6270  PRINT "570 H = L + I";: RETURN

6280  PRINT "580 IF A$ (L) < = A$
   (H) THEN 640";: RETURN
6290  PRINT "590 T$ = A$ (L)";: RETURN
6300  PRINT "600 A$ (L) = A$ (H)"
   ;: RETURN
6310  PRINT "610 A$ (H) = T$";: RETURN

6320  PRINT "620 L = L - I";: RETURN

6330  PRINT "630 IF L > 0 THEN 57
   0";: RETURN
6340  PRINT "640 P = P + 1";: RETURN

6350  PRINT "650 IF P > F THEN 52
   0";: RETURN
6360  PRINT "660 GOTO 560";: RETURN

6400  REM  === LINE COMMENTS ===
6410  PRINT "  SET <I> TO TOTAL I
   TEMS IN ARRAY";: RETURN
6420  PRINT "<I>/2 IS THE INTERVA
   L BTW'N ITEMS";: RETURN
6430  PRINT "IF INTERVAL=0,SORT I
   S COMPLETE";: RETURN
6440  PRINT "FLAG SIGNALS END OF
   THIS SORT PASS";: RETURN
6450  PRINT "RESET POINTER TO STA
   RT OF ARRAY";: RETURN
6460  PRINT "LOW SUBSCRIPT SET =
   TO POINTER";: RETURN
6470  PRINT "SETUP # OF PLACES BT
   W'N ITEMS";: RETURN
6480  PRINT "COMPARE LOW/HIGH ITE
   MS IN ARRAY";: RETURN
6490  PRINT "COPY LOW ITEM INTO T
   EMP STRING";: RETURN
6500  PRINT "COPY HIGH ITEM INTO
   LOW ITEM";: RETURN
6510  PRINT "COPY TEMP STRING INT
   O HIGH ITEM";: RETURN
6520  PRINT "SUBTRACT INTERVAL FR
   OM LOW SUBSCRIPT";: RETURN
6530  PRINT "IF <L> MORE THAN 0 T
   RY TO MOVE LOWER";: RETURN
6540  IF SW$ = "SWAP YES" THEN  PRINT
   "P LESS THAN 0-MOVE TO NEXT
   HIGHER ITEM";: RETURN
6545  PRINT "NO EXCHANGE-MOVE TO
   NEXT HIGHER ITEM";: RETURN
6550  PRINT "IF POINTER PAST FLAG
   ,THIS PASS DONE";: RETURN
6560  PRINT "PASS NOT COMPLETE,CO
   MPARE NEXT PAIR";: RETURN
6700  REM  === STORE RESULTS OF P
   ASS ===
6710  FOR X = 1 TO 26:D$(D,X) = A
   $(X): NEXT
6720  RETURN
6800  REM  === UPDATE PTRS AND AR
   RAY ===
6810  INVERSE : VTAB 19: HTAB 2: PRINT
   S1$;: VTAB 17: HTAB 2: PRINT
   S1$;: IF SC > 1 THEN  VTAB 1
   8: HTAB 6 + LX: PRINT A$(LX)
   ;: HTAB 6 + HX: PRINT A$(HX)
   ;
```

```
6815 VTAB 17: HTAB 6 + L: PRINT
   A$(L);: HTAB 6 + H: PRINT A$
   (H);: VTAB 18: HTAB 6 + L: PRINT
   " ";: HTAB 6 + H: PRINT " ";
6820 VTAB 19: HTAB 2 - LEN ( STR$
   (L)) + L: PRINT "A$(" + STR$
   (L) + ")^";: HTAB 6 + H: PRINT
   "^A$(" + STR$ (H) + ")";
6830 VTAB 16: HTAB 3: PRINT SW$;
   : NORMAL
6840 VTAB 7: HTAB HT + 1: PRINT
   A$(L);
6850 VTAB 8: HTAB HT + 1: PRINT
   A$(H);
6860 LX = L:HX = H: RETURN
6900 REM --- RESULTS TO SCREEN
   ---
6910 FLASH : VTAB 22: HTAB 6: PRINT
   " PUSH <RETURN> FOR FINAL DA
   TA";: GET I$
6920 HOME : INVERSE : VTAB 4: HTAB
   5: PRINT "   ORIGINAL UNS
   ORTED ARRAY   ": VTAB 5: HTAB
   5: PRINT "    ";: FOR X = 1
   TO 26: PRINT D$(1,X);: NEXT
   : PRINT " ";
6930 VTAB 6: HTAB 5: PRINT "PASS
      ";: PRINT "(RESULTS OF
   PASS)   ";
6940 FOR X = 2 TO 5: VTAB 5 + X:
   HTAB 5: PRINT " " + STR$ (
   X - 1) + "   ";: FOR Y = 1 TO
   26: PRINT D$(X,Y);: NEXT Y: PRINT
   "  ";: NEXT X
6950 VTAB 11: HTAB 5: PRINT "
    (FINAL ORDER OF ARRAY)
    ";
6960 VTAB 15: HTAB 3: PRINT "TOT
   AL # OF STATEMENTS EXECUTED
   = " + RIGHT$ ("   " + STR$
   (RS),3);
6970 VTAB 16: HTAB 3: PRINT "TOT
   AL # OF COMPARISONS MADE
   = " + RIGHT$ ("   " + STR$
   (RC),3);
6980 VTAB 17: HTAB 3: PRINT "TOT
   AL # OF EXCHANGES MADE
   = " + RIGHT$ ("   " + STR$
   (RE),3);
6990 FLASH : VTAB 20: HTAB 5: PRINT
   " PUSH <RETURN> FOR MAIN MEN
   U";: GET I$: NORMAL : RETURN
7000 REM === SUPERVISE NON-TRAN
   S ===
7010 CLEAR : GOSUB 8100
7020 GOSUB 8200
7030 GOSUB 7200
7040 GOTO 1100
7200 REM --- NTS CONTROL ---
7210 HOME : INVERSE
7220 VTAB 2: HTAB 6: PRINT " >
   >> REAL-TIME SORT <<<   "
7230 VTAB 4: HTAB 6: PRINT "   O
   RIGINAL UNSORTED ARRAY ";: VTAB
   5: HTAB 6: PRINT " ";: GOSUB
   9000: PRINT " ";
7240 VTAB 10: HTAB 6: FLASH : PRINT
   " START SORT - PUSH <RETURN>
   ";: GET I$: VTAB 10: HTAB 6:
   INVERSE : PRINT " START SOR
   T - PUSH <RETURN> ";
7250 VTAB 9: HTAB 6: PRINT " ***
   *     STARTED     **** ";
7260 GOSUB 7500: VTAB 8: HTAB 6:
   PRINT " ****      FINISHED
     **** ";
7270 VTAB 6: HTAB 6: PRINT " ";:
   GOSUB 9000: PRINT " ";
7280 VTAB 7: HTAB 6: PRINT "
   FINAL SORTED ARRAY    ";
7290 FLASH : VTAB 12: HTAB 6: PRINT
   " PUSH <RETURN> FOR MENU
   ";: GET I$
7300 NORMAL : RETURN
7500 REM === NON TRANS SORT ===
7510 I = 26
7520 I = INT (I / 2)
7530 IF I = 0 THEN RETURN
7540 F = 26 - I
7550 P = 1
7560 L = P
7570 H = L + I
7580 IF A$(L) < = A$(H) THEN 76
   40
7590 T$ = A$(L)
7600 A$(L) = A$(H)
7610 A$(H) = T$
7620 L = L - I
7630 IF L > 0 THEN 7570
7640 P = P + 1
7650 IF P > F THEN 7520
7660 GOTO 7560
7670 RETURN
8100 REM === INIT VARIABLES ===
8110 DIM A$(26),B$(26),C$(26),D$
   (5,26)
8120 A$ = "ABCDEFGHIJKLMNOPQRSTUV
   WXYZ"
8130 S1$ = "                 "
8140 TR$ = ">>> THE TRANSPARENT S
   ORT <<<"
8150 I = 0:F = 0:P = 0:L = 0:H =
   0:RS = 0:RC = 0:RE = 0:D = 0
   :SC = 3:S = SC ^ 2:SH$ = "":
   T$ = "": FOR X = 1 TO 26:A$(
   X) = "": NEXT
8160 HT = 37:TH = 35
8170 RETURN
8200 REM === MENU FOR ALPHA SEL
   ECT ===
```

```
8210 TI$ = ">>> ALPHABET MENU <<<
     "
8220 C = 6:C$(1) = "IN ORDER.....
     .. ALPHABET":C$(2) = "REVERS
     ED ORDER ALPHABET":C$(3) = "
     SCRAMBLED ORDER ALPHABET":C$
     (4) = "RANDOM LETTERS ALPHAB
     ET":C$(5) = "SELECT YOUR OWN
     ALPHABET":C$(6) = "RETURN T
     O MAIN MENU"
8230  GOSUB 9200: IF K = 6 THEN 1
     100
8240  ON K GOSUB 8300,8400,8500,8
     600,8700
8250  RETURN
8300  REM  === ALPHABET INTO ARRA
     Y ===
8310  FOR X = 1 TO 26:A$(X) =  MID$
     (A$,X,1): NEXT
8320  RETURN
8400  REM  === REVERSE ALPHABET =
     ==
8410  GOSUB 8300: FOR X = 1 TO 13
     :TE$ = A$(X)
8420 A$(X) = A$(27 - X):A$(27 - X
     ) = TE$: NEXT
8430  RETURN
8500  REM  === SCRAMBLED ALPHABET
     ===
8510  GOSUB 8300
8520  FOR X = 1 TO 26:Y =  INT ( RND
     (1) * (27 - X)) + X:TE$ = A$
     (X)A$(X) = A$(Y):A$(Y) = TE
     $
8530  NEXT X: RETURN
8600  REM  === RANDOM LETTERS ===
8610  FOR X = 1 TO 26:A$(X) = CHR$
     ( INT ( RND (1) * 26) + 65):
     NEXT
8620  RETURN
8700  REM  === USER SELECT ALPHAB
     ET ===
8710  HOME : GOSUB 8300:P1 = 1
8720  VTAB 2: HTAB 6: PRINT "SELE
     CT ALPHABET TO BE SORTED";: VTAB
     3: HTAB 6: PRINT "-----------
     -------------------";: VTAB 4
     : HTAB 6: PRINT "PUSH <RETUR
     N> WHEN COMPLETE";
8730  INVERSE : VTAB 10: HTAB 7: GOSUB
     9000: NORMAL : VTAB 15: HTAB
     7: PRINT "LEFT/RIGHT ARROWS
     TO MOVE";: VTAB 17: HTAB 7: PRINT
     "PUSH ANY LETTER TO CHANGE";
     : VTAB 19: HTAB 3: PRINT "(S
     PACE-BAR) TO RETURN TO MAIN
     MENU)";
8740  FLASH : VTAB 10: HTAB 6 + P
     1: PRINT A$(P1);: VTAB 11: HTAB
     6 + P1: PRINT "^";: VTAB 12:
     HTAB 5 + P1: PRINT " ^";: NORMAL
8750  GET I$: IF I$ =  CHR$ (13) THEN
     8810
8760  IF I$ =  CHR$ (32) THEN 110
     0
8770  IF I$ =  > "A" AND I$ <  =
     "Z" THEN A$(P1) = I$: FLASH
     : VTAB 10: HTAB 6 + P1: PRINT
     A$(P1);: VTAB 12: HTAB 5 + P
     1: PRINT " ^";: NORMAL : GOTO
     8750
8780  IF I$ =  CHR$ (8) AND P1 >
     1 THEN  INVERSE : VTAB 10: HTAB
     6 + P1: PRINT A$(P1);: NORMAL
     : VTAB 11: HTAB 6 + P1: PRINT
     " ";: VTAB 12: HTAB 5 + P1: PRINT
     "  ";: FLASH :P1 = P1 - 1: VTAB
     10: HTAB 6 + P1: PRINT A$(P1
     );: VTAB 11: HTAB 6 + P1: PRINT
     "^";: VTAB 12: HTAB 5 + P1: PRINT
     " ^";: NORMAL : GOTO 8750
8790  IF I$ =  CHR$ (21) AND P1 <
     26 THEN  INVERSE : VTAB 10: HTAB
     6 + P1: PRINT A$(P1);: NORMAL
     : VTAB 11: HTAB 6 + P1: PRINT
     " ";: VTAB 12: HTAB 5 + P1: PRINT
     "  ";: FLASH :P1 = P1 + 1: VTAB
     10: HTAB 6 + P1: PRINT A$(P1
     );: VTAB 11: HTAB 6 + P1: PRINT
     "^";: VTAB 12: HTAB 5 + P1: PRINT
     " ^";: NORMAL : GOTO 8750
8800  GOTO 8750
8810  RETURN
9000  REM  === DISPLAY ARRAY ===
9010  FOR X = 1 TO 26: PRINT A$(X
     );: NEXT
9020  RETURN
9200  REM  === MENU SELECTOR ===
9210  HOME
9220  HTAB 10: VTAB 2: PRINT TI$
9230  FOR X = 1 TO C: VTAB (X * 2
     + 5): HTAB 10: PRINT C$(X):
     NEXT
9240  VTAB 20: HTAB 5: PRINT "LEF
     T/RIGHT ARROWS FOR SELECTION
     ";: VTAB 22: HTAB 6: PRINT "
     PUSH <RETURN> FOR YOUR CHOIC
     E";
9250 K = 1: VTAB (K * 2 + 5): HTAB
     10: FLASH : PRINT C$(1);
9260  GET I$:I =  ASC (I$): IF I <
     > 13 AND I <  > 8 AND I <  >
     21 THEN 9260
9270  IF I = 13 THEN  GOTO 9320
9280  VTAB (K * 2 + 5): HTAB 10: NORMAL
     : PRINT C$(K);
9290  IF I = 8 THEN K = K - 1: IF
     K < 1 THEN K = C
9300  IF I = 21 THEN K = K + 1: IF
     K > C THEN K = 1
9310  VTAB (K * 2 + 5): HTAB 10: FLASH
     : PRINT C$(K);: GOTO 9260
9320  NORMAL :I = 0: RETURN
```

最新遊戲軟件評價

·FLOPPY
·RAINY DAY FUN
·SWASH BUCKLER II
·BUCKAROD BANZAI

李德華

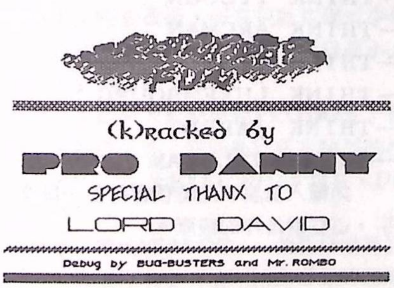
編者註：我們歡迎讀者投稿介紹各類新舊不同的遊戲軟件玩法、破解法。投稿辦法簡單，可參閱投稿需知。所有被介紹的均必須附同該磁碟，我們會將磁碟退回。稿件一經刊出即有稿酬。

RAINY DAY FUN是一個適合五至八歲的遊戲，這遊戲最大特色是能夠用PRINTER 來輔助遊戲，而這遊戲甚至可利用MOUSE來控制。

RAINY DAY FUN 共分十個遊戲。分別是：

一、摺飛機遊戲

在這遊戲中，電腦提供不同飛機的模式，選擇完畢後，使用你的控制器來控制摺紙的方向，直至完成爲止。



二、聯線

在螢幕上，有無數的數字在它們所屬的點上，把它們連接便成一幅美麗的圖畫。

三、驢的跑動

這個只是看，並沒有得玩。

四、印生日卡、賀卡

這個可算是最有用的了，首先繪畫喜歡的圖案或從磁碟中LOAD出（自動），再把一些預置的圖案加插上畫中，再打入姓名，收卡人的名字等等便成。

五、BANNER

印海報和印生日卡的工作模式差不多，在此不再詳談。

六、迷宮遊戲

這遊戲的主角就是你（老鼠），你要設法找到出口，當完成後，便開始第二版。

七、砌圖遊戲

開始時，電腦會SHOW一幅圖畫給你看，待二至三秒後，便由電腦分十六格來搞亂，能否還原就看你的IQ有多高了。

八、畫圖

首先玩者會在螢幕上顯示一個老人的樣貌，把指標向上移，那麼指標便會變成一枝筆，按SPACE一下，便能夠繪畫，再按SPACE便不畫或作ERAS—E 用途。

九、塡顏色

RAINY DAY FUN 預置了十多幅以線繪畫的圖畫，選擇適當的圖畫後，便再選擇顏色，然後便能上顏色。

十、水晶球

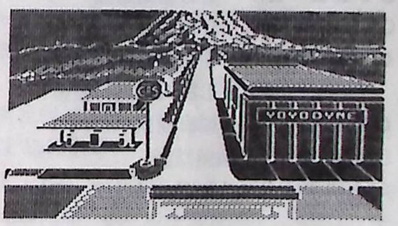
當BOOT起這遊戲的時候，首先看見巫師走到水晶球旁，待一會兒，便會看見一條關於星星的問題，於是你便要發問一條類似的問題，但當然不要以較高深的問題來詢問，否則往往有出人意表的答案。

RAINY DAY FUN 除了印卡及海報適合較大年紀的人玩外，其餘只適合五至八歲的小童娛樂，所以不要要求太高。

又筆者得知RAINY DAY FUN不乏BUG版本，但多數是"驢的跑動"較爲有BUG，所以最好還是COPY原版 本爲佳。



遊戲名稱：RAINY DAY FUN
遊戲類型：適合五至八歲小童啓發智能遊戲。
評　　分：＊＊＊

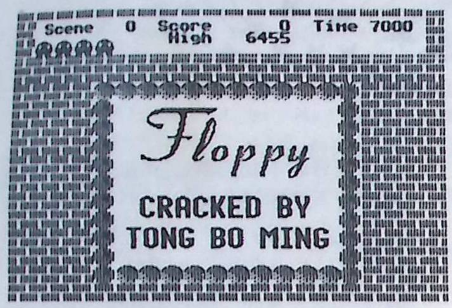


BUCKAROD BANZAI是蜘蛛俠的同廠出品。它比蜘蛛俠更加難玩，是一個ADVENTURE遊戲。

到目前爲止，筆者仍然不知道這故事的內容，但我大致上也知道玩法，例如要進入地方（例如HOUSE），那麼便要打入ENTER HOUSE ，而不是GO HOUSE ，筆者經過一番研究後，發覺高登大部份的版本是有BUG的，所以如果不幸手上是有這版本的讀者，也無可怨言了。

由於上述原因，筆者只能夠玩上半部份，最後目的地是把RADIO及AN—TENNA BATTERY等接合，而且筆者更得到一條FORMUIA ，說要把一些東西混合，才能開動那架吉普車，如果日後有機會的話，便會再和大家分享研究的成果。

遊戲名稱：BUCKAROD BANZAR
遊戲類型：蜘蛛俠同廠出品
ADVENTURE
評　　分：暫不置評



FLOPPY 除了玩之外，更可自製迷宮，由於製造迷宮的方法比玩法困難得多，所以先講這方面的問題。首先簡單介紹FLOPPY 控制的方法：

CTRL—B　重新開始遊戲
CTRL—R　重新開始遊戲
CTRL—D　DEMO
CTRL—A　自我毀滅但仍然在這LEVEL進行遊戲。
CTRL—E　編輯模式

編輯模式共分多個鍵控制
向上向下共分：I，K，A，Z，P，+
向左向右共分：J，L，⟵，⟶
至於標題輸入方面，可按RETURN 進行
按<，>選擇編輯的版圖

CTRL—L　LOAD LEVEL
CTRL—S　SAVE LEVEL
CTRL—P，Q，N　離開模式
CTRL—C　清洗畫面
1—8 選擇牆—石頭—怪獸等物體

至於編輯LEVEL方面，其實一看編輯的畫面也意會到，首先清洗畫面，然後把游標移到適當的地方，選擇適當的物件放置，跟着設置石頭、怪獸、鐵球等東西，最後便是要設置終點了，跟着按RETURN 打入名字，按CTRL—S把它SAVE 起來。

言歸正傳，現介紹遊戲的玩法，玩法方面是要把鐵球推落終點，期間避過怪獸等襲擊，而你的唯一武器就是擲仙人掌或把石頭推落而壓死怪獸。

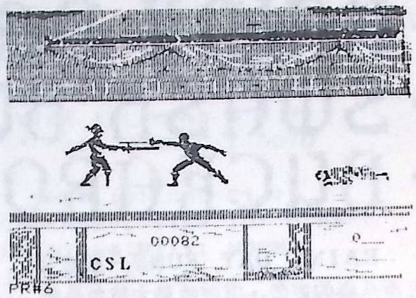
遊戲名稱：FLOPPY
遊戲類型：自製迷宮ACTION 遊戲
評　分：＊ ＊

SWASH BUCKLER II 是SW—ASHBUCKLER I 的續集，雖然是續集，但看上去並沒有多大分別，除了背景及人物變動外，所有的玩法跟以往一樣，並沒有多大的新意。

至於KEYBOARD 的控制大致如下：

A—向左行　J—中劍
D—向右行　M—下劍
S—轉身　L—飛身劍
I—上劍

總括來說，如不是要求太高的話，也值得一玩。



遊戲名稱：SWA SHBUCKLER II
遊戲類型：ACTION
評　分：＊ ＊

MINDSHADOW

MINDSHADOW的故事是講及你是失憶者，因為一次意外而飄流到荒島上，因此你便要設法離開荒島。在島上有STEEL—VINE—MAP 等的東西，小心把它們存起，對於日後有所幫助。閒話休說，遊戲的目的是要令自己的記憶力恢復，現在筆者提供最快的破解方法給大家。只要依着輸入便行。

—THINK TYCOON
—THINK ARCMAN
—THINK BOB
—THINK LUXEMBOURG
—THINK JARED
—THINK WILLIAM

如輸入無誤，此時便會打出一段文字，這些就是你的經歷。

AFC 收集所有熱門遊戲軟件供免費借用

遊戲軟件評論專欄每期所介紹的遊戲軟件均收藏在APPLE FAN CLU—B 內的軟件庫供會員借用，軟件庫內收集有目前市面最流行熱門的軟件，加入APPLE FAN CLUB，閣下可完全免費借用所有這些軟件，包括遊戲及UTILITY 查詢。參加APPLE FA—N CLUB 方法，可來信九龍中央郵政信箱71193號，連同一個貼上五角郵票的回郵信封及一個一元郵票，信封面請註明「索取AFC入會表格」。

加添一個獨特的指令 BCOPY

趨向完美的 DOS

楊德信

DOS 3·3 的缺點，相信大家在使用時都會發覺得到。它的改良方法已有很多文章討論過，大多數都是速度方面的改良，和40TRACK 的擴展，而其他的缺點却忽略了。

因此本文就把一些被人忽略了而又在使用時覺得厭煩的缺點改良過來，而且還加上了一個指令，現將改良列表如下：

1.使非直接指令能夠直接使用
2.以〝—〞代替RUN，BRUN 及EXEC
3.CATALOG 長目錄時，按CTRL－C可停止執行
4.修改BSAVE 長度使能處理大過＄7FFF 的資料
5.能夠INIT 40 TRACK
6.錯誤訊息處理的改良
7.BCOPY 指令的加入

大家看完上表後，相信會對今次的修改大感興趣，現將各點詳述如下：

項目1：經修改後，所有只能在程式中使用的指令現在都可以由鍵盤直接鍵入使用。不過要小心一點，以免把檔內的資料弄亂了。

項目2：如果大家使用過PRODOS，都不會忘記〝—〞指令的優越性，它能執行所有類型的檔，而不須打入RUN，BRUN 及EXEC 指令。現在，DOS 3·3 也可以做到了。

項目3：當CATALOG 一個長目錄尋找自己所需要的檔時，却發現所要的檔剛剛在目錄的開頭或中間，那只好牢記檔名待CATALOG 完後才作處理，或立即按RESET 來停止執行。用第一個方法是非常之麻煩，而第二個方法却很容易弄壞磁碟，所以經過這項修改後，每當列完一頁目錄後等待按鍵時，可按CTRL－C 停止，或其他鍵繼續列印。

項目4：經過第四項修改後，很長的檔現在可以BSAVE 入碟了。

項目5：INIT 40 TRACK雖然有很多文章介紹過，但這些都是要執行一個程式才能達成任務。但經這項修改後，DOS 會自動INIT 40 TRACK 和自動釋放TRACK 35－39及TRACK 2 SECTOR 5－15 的空間。

項目6：每當錯誤發生，通常是打錯字，因此又要把敍述重打一次，相當費時，所以聰明的朋友會利用APPLE 的ESC 功能來把剛才打入的敍述再TRACE 一次及作修改，不過仍覺麻煩。經修改後，每當錯誤發生，游標會自動反回剛才輸入敍述的第一個字元上，所以大家可以輕而易擧的把敍述再TRACE一次及作修改。

此外，錯誤語句的列印也改變了，這是為了項目3所作的改變。當我們按CTRL－C 後，游標會在最底的一行，當錯誤發生時，如果未作修改的話，螢幕會上捲三行來印出錯誤語句，於是最頂部的三個檔名便不見了；但經修改後，螢幕只會上捲一行來印出語句，所以就只有第一個檔才被捲去。不過這是沒問題的，因爲如果大家細心研究一下的話，就會發覺第一個檔就是上一頁目錄的最後一個檔，所以可以省略之；但如果這頁是第一頁目錄又怎樣呢？我相信，大家不會忘記DISK VOLUME 的存在。

項目7：除以上修改外，本項目特別加入了一個新指令——BCOPY。這個指令能夠把一個二進制檔從一隻碟抄到另一隻碟上，它首先叫用BLOAD 副常式來載入檔案，然後再叫用BSAVE 副常式來寫到另一隻碟上，所以檔案的位置不能與DOS 重叠，否則會產生不可預料的後果。幸好的，這個指令提供了A這個鍵語，它的功用和在BLOAD的一樣，指令會先把檔案抄到A或A＄所指的地址上，然後才把檔案由這地址抄到另一隻碟上，所以BLOAD或〝—〞程式時，要再用A鍵語來載回原本的地址。現擧例說明這指令的用法：

首先準備兩片INIT 過的空碟

打入CALL －151

```
300:A2 00 BD 0F 03 F0 07 20
    ED FD E8 4C 02 03 00 8D
    C8 C5 CC CC CF A1 00
```

放入第一片空碟

打入BSAVE ABC，A＄300，L＄18

待磁碟機停後打入—ABC 來執行程式，螢幕會印出HELLO！字樣。

打入BCOPY ABC，接着DOS 便載入程式到＄300

接着有一游標在檔名後閃動。

換入第二片空碟並任意按鍵。

待寫入完畢後打入CATALOG 這便看到ABC檔在第二片磁碟上，試着執行它，結果一樣。

接着打入DELETE ABC

打入300：00 N 301＜300，318M

換入第一片磁碟並打入BCOPY ABC，A＄1000 或打入BCOPY ABC，A 4096

磁碟機亮着並載入程式到＄1000起始的地址。

接着游標出現

換入第二片磁碟及按鍵寫入完畢後試着執行，但錯誤發生，因爲程式現在位於＄1000而不是＄300現在試打入—ABC，A＄300，結果成功執行。

至此，各位應該熟識各功能，跟着就是修改的步驟：BOOT 起DOS 3·3 或市面上經修改的快速DOS，但DAVID DOS 及PRONTO DOS 不能使用，因爲副程式進入點不同。

順序打入：

```
A021:4C 32 A0
```

跳過檢查非直接指令副程式

```
9D24:2B AA
A890:AD
A910:75
```

修改指令表，鍵語表及命令處理常式進入表以刪去BR−UN，EXEC及RUN指令並加入"−"指令及指向PAT−CH RUN副常式

```
AE39:20 F0 A4
```

修改CATALOG 副常式以指向PATCH CAT 副常式

```
A964:FF
```

修改BSAVE 長度使能處理達到$FFFF

```
AEB5:A0
AEB3:08
BEFE:28
AECC:20 24 A0
```

改變INIT功能處理常式及RTWS之FORMAT 程式，使能夠INIT 40 TRACK及釋放TRACK 35−39之空間，指向PATCH INIT 副常式來釋放TRACK 2 SECTOR 5−F之空間並改變VTOC之磁軌數目爲$27

```
A972:87
```

改變印出錯誤語句所需的行數

```
A6EC:20 4F 9D
```

指向PATCH ERROR 副常式

```
9D52<9D54.9D55M
9D44<9D48.9D55M
9D32<9D34.9D55M
9D26<9D28.9D55M
A8FE<A902.A908M
A8E1<A8E7.A908M
A8B1<A8B5.A908M
A891<A898.A908M
A93D<A93F.A940M
A92F<A933.A940M
A91D<A91F.A940M
A911<A913.A940M
```

刪除IN#，PR#及CHAIN 命令以放出空間給修補副常式用

```
AA2C:20 29 A2 0A F0 0A C9 08
     D0 03 4C 8E A3 4C D1 A4
     4C C6 A5
```

PATCH RUN 副常式

```
A4F0:20 0C FD C9 83 D0 03 4C
     D0 03 60
```

PATCH CAT 副常式

```
A024:A9 E0 8D FC B3 AD FE BE
     8D EF B3 4C FB AF
```

PATCH INIT 副常式

```
9D4F:A5 25 E9 01 4C 5B FB
```

PATCH ERROR 副常式

```
9D4C:F8 A8
A8F3:42 43 4F 50 D9 00
A937:21 71
```

修改指令表，鍵語表及命令處理常式進入表以加入BCO−PY 指令及指向其副常式

```
A8F9:20 5D A3 A2 02 BD 5F AA
     9D 6B AA CA D0 F7 F0 30
A939:20 0C FD 4C 3D A3
```

BCOPY 命令執行副常式

INIT 一隻空碟

此後，這張碟就可以永遠存有修改過的DOS，大家可以BO−OT這張碟來載入修改過的DOS及用來INIT 其他的碟。如果執行者有問題，這便是所用的DOS不適合用來修改。

最後要提一提大家的是：因爲要提取空間供副常式存放，所以在此把DOS中的PR#，IN#及CHAIN指令刪除，雖然DOS 3·3也有一些空間給日後修改之用，但爲了兼顧一些快速DOS已佔用了這些地址（至少本人所用的正是），所以只能刪除一些少用的指令。其中CHAIN 指令可說是很少用到的指令，甚至會給人遺忘了，而PR#及IN#兩指令，因爲APPLESOFT 翻譯器已提供，所以也刪除掉；在程式中，大家可以用PR#來代替PRINT CHR$(4)"PR#"，而IN# 也是一樣。另外，本修改只適合48K DOS，而RAM CARD DOS 不能使用。

至此，本文也告一段落，日後再有更好的修改，本人定會向大家介紹。■

電腦時代
分類廣告
查詢詳情
請電：3-312007

把機械碼化成BASIC

JANSEN

各位喜愛寫BASIC的朋友，有沒有覺得在你的程式上加上機械語（例如SHAPE TABLE，音樂程式等）是一件十分麻煩的事。如果利用BLOAD來加在程式內，不但程式不能一氣呵成，而且程式每次進行時又要來一輪磁碟機的操作，不是太費時失事嗎？

現在要介紹一個能間接把機械碼化成BASIC的程式。

程式的原理是把磁碟上的BINARY FILE載入，然後根據每一個地址的數值，變成一個DATA 放在一個可EX－EC 的TEXT FILE裏。當EXEC 這個TEXT FI－LE ，就會產生一個POKE DATA的BASIC PROG－RAM 。每當RUN 這個PROGRAM ，就會把適當的數值POKE到適當的地址上了。

這程式開始時，電腦會詢問那個想要變成BASIC的名稱。然後電腦會把這FILE 載入電腦，接着電腦會問你開始的行號，這行號是那BASIC PROGRAM 的第一行。跟着一輪的磁碟機操作後，你可以在那磁碟中找到一個.T尾的TEXT FILE ，載入你希望連結的程式，然後EXEC 這.T尾的FILE，在適當地方加上GOSUB，RETURN，現在你成功了，你已把一段機械碼加在你的程式內了。■

```
1   REM   **************
2   REM   * BIN-->BAS *
3   REM   *    BY     *
4   REM   *  JANSEN   *
5   REM   **************
10 D$ =  CHR$ (4)
20  TEXT : HOME
30  PRINT "INPUT THE BINARY FILE
    NAME:"
40  INPUT "";PE$
50  PRINT : PRINT D$;"BLOAD ";PE$

60 AD =  PEEK ( - 21902) +  PEEK
    ( - 21901) * 256
70 LE =  PEEK ( - 21920) +  PEEK
    ( - 21919) * 256
80  PRINT "THE ADDRESSES YOU JUST
    BLOAD ARE"
90  PRINT "  A";AD;",L";LE
100 PE$ = PE$ + ".T"
110  PRINT : PRINT "THE TEXT FILE
    NAME :": PRINT PE$
120  PRINT
130  PRINT "INPUT THE STARTING LI
    NE NUMBER :"
140  INPUT "";LN
150  PRINT D$"MONCIO"
160  PRINT D$"OPEN ";PE$
170  PRINT D$"WRITE ";PE$
180  PRINT LN;" FOR I = ";AD;" TO
    ";AD + LE;" : READ DT : POK
    E I,DT : NEXT I"
190 LN = LN + 10
200  PRINT LN;"DATA";:LN = LN + 1

210  FOR I = ADT TO AD + LE
220 PK =  PEEK (I)
230  PRINT PK;
240  IF (AD - I) / 50 =  INT ((AD
    - I) / 50) THEN  PRINT : PRINT
    LN;"DATA ";:LN = LN + 1: GOTO
    270
250  IF I = LE + AD THEN 270
260  PRINT ",";
270  NEXT I
280  PRINT
290  PRINT D$;"CLOSE ";PE$
300  PRINT D$;"NOMONCIO"
```

更　　正

第十八期P.77〔步步爲營—TRIP—PLES〕一文APPLESOFT 程式列表中的行句430 應爲：

```
430  DATA 156,247,134,467,345,457
     ,458,367,138,146,136,678,358
     ,125,248,148,568,178,567,127
     ,578,268,236,137,258,347,167
     ,237,234,378,267,346,478,000
     ,BC,CC,CB,CA,BA,AA,AB,AC
```

編 輯 部 小 啟

各位熱心讀者寄來的投稿，我們經全部收到並安排在各期刊出，如未見有回音，切勿心急及灰心，誤會稿件未被接納。請耐心等候，同時希望再接再勵，多寫多作程式。落選稿件除一行程式、捉虫外，我們會退回原稿。一行程式或捉虫訓練稿件如要退稿，請附回郵信封。

各位作者如見到自己作品經已登出，請於該期出版一星期後，前來讀者服務部領取稿酬。

APPLE FAN CLUB NEWSLETTERS

通訊欄

加入全港最大的蘋果電腦用家組織。閣下可以用最低廉的代價借取到無數的外國原裝軟件。

請附同一個貼上五角郵票的回郵信封及另外一個一元郵票，寄來九龍中央郵政信箱 71193 號電腦時代出版社收。信封面請註明「索取AFC章程及入會申請表」。

在上幾期會訊中，我曾提及利用 ULTRA FAST COPY 來 COPY PENGUIN 所出品的 SOFTWARE，我想大家都修改成功，但大家有沒有在 COPY 途中，有曾想過將這間公司的產品 DEPROTECT，即將他們改造成 COPYA 版本呢，如果有的話，我便將此方法介紹給大家。

首先 BLOAD PENGUIN COPY，然後修改：

1321：EA EA EA
1339：EA EA EA
816：DE
81C：DE

修改完後所製造出來的便是 COPYA 版本，但還有一點要注意的是此製成品的 DOS，即 TRACK＄00-02還是原來的 PROTECTED DOS，所以我們必須修改回 NORMAL DOS 才行，只要利用 COPY Ⅱ PLUS 之 UTILITY 來 COPY NORMAL DOS 便可，但有一點要注意的是，此方法並不能使全部 PENGUIN 所出品的 SOFTWARE 改良爲 COPYA 版本的。

```
07FD- 20 58 FC
0800- A9 20 8D 72 03 A9 00 85
0808- F4 85 F5 A9 10 8D 89 03
0810- A9 01 8D 8C 03 8D 82 03
0818- A5 F4 8D 84 03 A5 F5 8D
0820- 85 03 20 B5 08 20 26 03
0828- A9 10 8D 89 03 A9 02 8D
0830- 8C 03 8D 82 03 A5 F4 8D
0838- 84 03 A5 F5 8D 85 03 20
0840- A0 08 20 26 03 E6 F4 E6
0848- F4 A5 F4 C9 24 F0 03 4C
0850- 0B 08 A9 01 85 F4 A9 10
0858- 8D 89 03 A9 01 8D 8C 03
0860- 8D 82 03 A5 F4 8D 84 03
0868- A5 F5 8D 85 03 20 B0 08
0870- 20 26 03 A9 10 8D 89 03
0878- A9 02 8D 8C 03 8D 82 03
0880- A5 F4 8D 84 03 A5 F5 8D
0888- 85 03 20 A0 08 20 26 03
0890- E6 F4 E6 F4 A5 F4 C9 23
0898- F0 03 4C 56 08 60 19 B1
08A0- A9 DE 8D 35 B9 8D 91 B9
08A8- A9 D5 8D 55 B9 60 8E 5D
08B0- A9 D4 8D 55 B9 A9 DA 8D
08B8- 35 B9 8D 91 B9 60 9D 0E
```

而由於可能有部份會友因沒有 ULTRA FAST COPY 的關係，所以我便利用刊登在電腦時代第19期中的活用 RWTS 的 RWTS 來加以修改成爲另一個可以將 PENGUIN 所出品的 SOFTWARE 爲 COPYA 版本。

首先要講明的是此程式必需要將 BACKUP 的磁碟先要 FORMAT 好，並且此程式只提供給與擁有兩部磁碟機的朋友，而只有一部磁碟機的朋友，只要修改此程式或以此程式作爲參考，加以改良亦可使用。

輸入程式後，BSAVE PENGUIN TO COPYA，A＄7FD，L＄C3。

使用此程式時，必須記着 FORMAT 3 BACKUP 磁碟，然後 BLOAD RWTS，A＄300，如果你的 DOS 不能使 RWTS 此程式直接讀入地址＄300時，只需要 BLOAD RWTS，A＄1300，然後鍵入300<1300．13FFM，最後才 BLOAD PENGUIN TO COPYA，當放好 PENGUIN 之 SOFTWARE 碟在 DRIVE 1及空碟在 DRIVE 2後，按 7FDG 便可以將整隻碟 COPY 好。

然後跟着做的是 COPY DOS 落 TRACK ＄00-02處，但這並不是完全成功的，最大而最煩的是要將整隻的 FILE TRACE，如當見到 JUMP SUBROUTINE 到＄A000-BFFF 任何一個位置，但除了 CALL BLOAD，BRUN 即＄A35D 或＄A38E 等外，全部皆改爲 EA，還有的便是在 APPLESOFT BASIC 所寫的 PROGRAM，如遇到 CALL 4000-5000任何地址等出現時，將它刪掉便大致成功，但可能會有部份的 PROGRAM 例外，修改後都不能成功，但以上的方法，不單只可以應用在 PENGUIN 的 SOFTWARE 身上，而且還可以應用在其他使用 NORMAL DOS PROTECT 的 PROGRAM 碟上，只要大家活用便可。

本欄資料由APPLE FAN CLUB提供。APPLE FAN CLUB（簡稱AFC是一個非牟利的蘋果電腦用家組織，每月定期出版會員通訊介紹最新軟件出版消息，遊戲軟件解拆保護的方法，BACKUP 軟件參數、更改遊戲難度、機會次數等。此外每月更有大量最新軟件／使用手冊免費借給會員。索取入會章程及資料請連同回郵信封另加付一元郵票寄九龍中央郵政信箱71193號。信封面請註明「索取AFC入會章程」。□

徵求一行及兩行程式設計有獎遊戲

自從徵求一行程式遊戲舉辦以來，獲得許多讀者的支持，每月均收到數以百計的作品，當然，部份有水準的作品我們經選登出來，至於未被接納的作者請勿灰心，繼續努力，多些編寫。

由於利用一行敍述句來寫出一個精彩的程式有一定的難度限制，未必能充份發揮個人的創作才華，爲此，我們再舉辦多一個遊戲——徵求二行程式設計，讓大家有二行敍述句的長度來發揮個人的才華。

二行程式設計比賽參加辦法及獎品和一行程式有獎遊戲一樣，但請在信封面註明二行程式。
各位讀者寫來程式時，必須用多行的形式寫來（參考本文的各位得獎例子），而且要書寫清楚。投稿請寄來CPO BOX 71193 。入選者可獲得任何一期的程式磁碟一張（請註明要的磁碟）。來稿信封請勿封口。 ■

以下程式是可帮助大家洗磁頭，程式開始後，你要輸入SLOT及DRIVE，然後程式會工作，磁頭會由磁軌零走到磁軌39，一共走四次，在工作期間，按ESC 可以停止。

李宗樑君可獲得第21期程式磁碟一張。

```
10    FOR I = 768 TO 773
    : READ A
    : POKE I,A
    : NEXT
    : HOME
    : INPUT "SLOT:";S
    : PRINT
    : INPUT "DRIVE:";D
    : POKE 47081,S * 16
    : POKE 47082,D
    : POKE 47092,0
    : FOR I = 1 TO 4
    : VTAB 5
    : PRINT "TIME=";I
    : FOR J = 0 TO 39
    : VTAB 7
    : PRINT "TK="J" "
    : POKE 47084,J
    : CALL 768
    : IF PEEK ( - 16384) < > 155
       THEN NEXT J,I
    : DATA 32,227,3,76,217,3
```

賴耀華君是首位獲得二行程式比賽的得獎者。賴君設計的是一個撞磚遊戲程式。用歡樂棒控制球板的左右。有聲响效果。遊戲目的是將磚牆撞穿，當彈球穿牆後，便完成一個LEVEL ，之後顯示所得分數，按BUTTON 0 繼續下一個LEVEL，每升一個LEVEL ，磚牆加厚一層，直至21層爲止。若球到達最底一行，則遊戲完結，分數會顯示在螢幕上。

賴君可贏得第六期程式磁碟一張。

```
1     GR
    : POKE 49234,0
    : CALL - 1998
    : FOR I = 0 TO L
    : COLOR= RND (1) * 12 + 2
    : HLIN 0,39 AT I * 2
    : NEXT
    : M = 20
    : N = 46
    : O = 1
    : P = - 1
    : FOR I = 0 TO 1
    : COLOR= 0
    : HLIN X,X + 4 AT 47
    : COLOR= 15
    : X = PDL (0) / 255 * 35
    : HLIN X,X + 4 AT 47
    : C = SCRN( M,N + P)
    : COLOR= 0
    : PLOT M,N + P
    : PLOT M,N
    : P = - P * (C > 0) + P * (C = 0)
8     Z = (M > 0 AND M < 39)
    : O = - O * NOT Z + O * Z
    : M = M + O
    : N = N + P
    : COLOR= 1
    : PLOT M,N
    : FOR J = 0 TO C
    : E = PEEK (49200)
    : NEXT
    : I = (N = 0 OR N = 47)
    : S = S + .05
    : NEXT
    : TEXT
    : HOME
    : PRINT "SCORE=" INT (S)
    : IF N = 0 THEN L = L + (L < 21)
    : S = S + L * 50
    : PRINT
    : PRINT "BONUS="L * 50
    : WAIT 49249,128
    : GOTO 1
```

馮兆民君寄來一個盛載磁碟的空套製造程式，讓大家郵寄磁碟時有一個適合大小吋的封套。

馮君可贏得第22期程式磁碟一張。

```
  PR# 1
: A$ = "+"
: B = 69
: FOR I = 12 TO 10 STEP - 2
: PRINT TAB( I)A$ TAB( B)A$
: FOR J = 1 TO 4
: PRINT
: NEXT
: B = 71
: NEXT
: PRINT TAB( 2)A$ TAB( 79)A$
: FOR J = 1 TO 29
: PRINT
: NEXT
: PRINT TAB( 2)A$ TAB( 79)A$
: PRINT
: PRINT TAB( 10)A$ TAB( B)A$
: FOR J = 1 TO 34
: PRINT
: NEXT
: PRINT TAB( 10)A$ TAB( B)A$
: PR# 0
```

馮南海君可取得第十七期程式磁碟一張。

本程式的目的，爲列印一順序文字檔案，其效果雖不甚特出，但有一項技巧却相當別緻，此爲其中所用到的PO－KE 指令，其目的爲模擬ONERR GOTO，因APPLESOFT 遇着ONE－RR GOTO，便會忽略以下以 "：" 分格的指令，而跳往下行，所以需要以POKE 模擬0 程式本身，能在INPUT SPEED 時控制列印速度，並以CTRL－S 暫停列印，而程式本身是在INPUT FILE NAME 時以RETURN 鍵輸入來完結，其中程式的行號，因會用ONE－RR GOTO 而不能改變。

```
1     D$ = CHR$ (4)
    : PRINT D$"CLOSE"
    : POKE 216,0
    : INPUT "FILE";F$
    : IF F$ < > "" THEN INPUT "SPEED=";
       S
    : POKE 244,0
    : POKE 245,1
    : POKE 246,0
    : POKE 247,1
    : POKE 216,128
    : SPEED= S
    : PRINT D$"OPEN"F$
    : PRINT D$"READ"F$
    : FOR I = 0 TO 1
    : INPUT A$
    : PRINT A$
    : I = 0
    : NEXT I
    : GET C$
    : GOTO 1
```

利用我們的分類廣告，閣下只要花極少的代價，便可將訊息傳到我們廣大的讀者網

無論是自己編寫的軟件、轉讓各類二手電腦及週邊、新出版書刊、代編程序服務、私人電腦科補習、程序員徵聘、待聘、軟件出讓等等，都可以利用電腦時代的分類廣告欄。

查詢詳情，請電：3-312007分類廣告部洽

印字機及監視器出讓

EPSON LX−80 只買了數月（仍有廠保用），因移民割讓$ 1100，綠色 12" 監視器$ 400，有意者請晚上10時半後電3-7262867 。

PRINTSHOP用多款造型加菲貓

十五款不同造型加菲貓專輯，專供PRINTSHOP使用，可供設計印製各種問候咭、生日咭、海報或橫額等。磁碟每張十五元。電腦時代讀者服務部出售，只限長期訂戶／AFC／MBC會員。可郵購。

TAKE Ⅰ製作電影

TAKE 1　電影製作比賽得獎作品——約會，每張十五元（連磁碟）電腦時代讀者服務部發售。

稱骨算命程式

中文稱骨算命程式，港人編寫，根據出生年、月、日、時計算個人一生運程，非常準確，每套25元，只售與MBC會員及本刊長期訂戶及ＡＦＣ會員。發售處：電腦時代讀者服務部。歡迎參觀示範。

港人創作ADV遊戲

港人創作ＡＤＶＥＮＴＵＲＥ遊戲軟件——地鐵風雲，由專欄作者李德華君創作，每個只售15元。本軟件並未加上任何保護，可隨意LIST 。只售與長期訂戶、ＡＦＣ會員及ＭＢＣ會員。讀者服務部發售。

CP／M用戶佳音

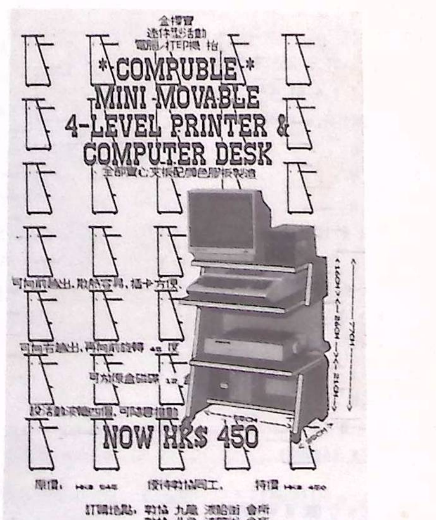
CP ／M用戶佳音，CP ／M　FORMAT 40（VERSI−ON Ⅱ），將CP／M格式磁碟FORMAT 成40軌，每套50元連使用手冊。電腦時代讀者服務部發售。

精工修理電腦／週邊咭

精工修理蘋果兼容及IBM 兼容電腦及週邊咭，取價公道，深水埗黃金商場地庫21號萬達電腦公司。3-610095

128K 咭特價

128K RAM咭特價發售，數量有限，售完即止　電腦時代讀者服務部3-312007



佔地少、實用性高迷你型活動電腦／打印機枱，全部採用實心夾板配顏色膠板製造，經濟實用。呎吋：77cm高×55cm濶×35cm深（底部）。

原價HK$ 545，現特價$ 490 發售。

代售處：電腦時代讀者服務部。

數量有限，售完即止。

優待教育專業人員協會會員：每部特價HK$450

教協會員可於佐敦道渡船街及北角渣華街會所訂購。

電腦時代精裝珍藏本 （1-12期合訂本）

隨書附送價值超過150元的購買程式磁碟優待券

精裝珍藏本，顧名思義，是將過去出版的電腦時代第1至12期所有的精彩內容滙集成一巨册，其中包括超過一百個的軟件程式使用說明、數十篇有關各類週邊硬件深入解剖報告、熱門軟件評介及用法文，是所有擁有APPLE或APPLE兼容電腦的用家必須珍藏的巨獻。

可能你已擁有第一至十二期，但這册精裝珍藏本依然有你值得擁有的價值，因爲我們在這册合訂本中加插了多篇從未刊登過的文章及程式，而且我們更隨書附送價值超過150元的購買程式磁碟優待券。

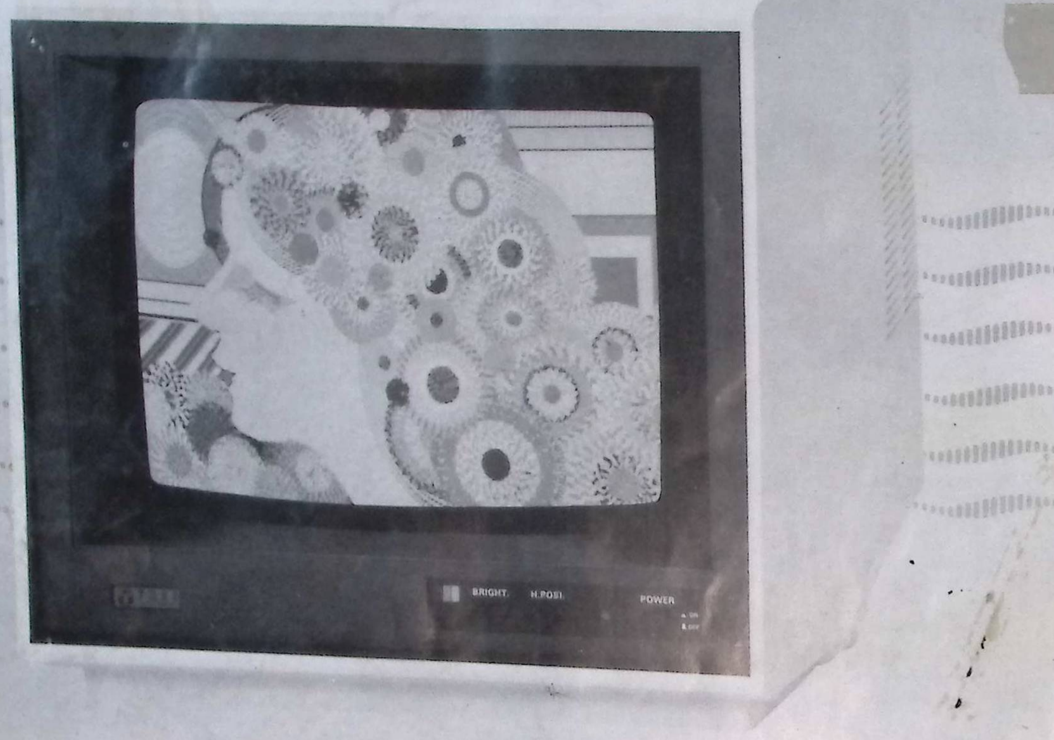
電腦時代精裝珍藏本全書厚度超過[illegible]，採用穩固的膠背釘裝。每册僅售50元。



TOEI Toei Electronics Co., Ltd.

TOEI系列彩色監視器

色彩更趨完美・影像更見逼眞



Compatible with IBM PC/PC jr., Apple IIc/IIe/III and other PCs



FTC系統規格表

Name of Model		FTC-1201-R	FTC-1203-R	FTC-1208-R	FTC-1203-HF	FTC-1208-HF	FTC-1208-W
Picture Tube		12", 90°Deflection, In-Line Type, P22, Hi-Contrast CRT (FTC-1203-R, FTC-1203-HF : Non-Glare)					
		0.47mm Dot Stripe Pitch	0.38mm Dot Trio Pitch	0.28mm Dot Trio Pitch	0.38mm Dot Trio Pitch	0.28mm Dot Trio Pitch	
Input Signal Level	RGB Video Signal	RGB Analog 2－5Vp-p (Positive), RGB plus Intensity TTL Level (Positive) or RGB TTL Level (Positive)			RGB plus Intensity TTL Level (Positive)		
	Sync. Signal	Composite or Separate (H. D, V.D) TTL Level (Positive or Negative)					
Scanning Frequency		H. 15.75kHz ±400Hz, V. 60Hz (50－65Hz)			H. 25kHz ±600Hz, V. 60Hz (50－65Hz)		H. 31.5kHz ±1kHz V. 60Hz (50－65Hz)
Video Bandwidth		20MHz－3dB		25MHz－3dB			
Characters to Display		2,000 Characters (80×25 Lines)			4,050 Characters (90×45 Lines) or 2,000 Characters (80×25 Lines)		7,200 Characters (120×60 Lines) (Ex)
Active Display Area		H. 220mm (Max. 240mm), V. 137mm (Max. 150mm)					
Active Display Time		H. 45μsec., V. 13.1msec.			H. 30.2μsec., V. 13.1msec.		H. 23.8μsec., V. 13.1msec.
Resolution		640Dots (FTC-1208-R : Max. 800Dots) ×200Lines (Max. 280Lines)			640 Dots ×400 Lines Max. 450 Lines	640 Dots/Max. 800 Dots ×400 Lines Max. 450 Lines	712 Dots/Max. 800 Dots ×512 Lines/Max. 550 Lines
Amplifier Type		Linear Amplifier					
Connector Type		8P Terminal					
Power Supply		AC 120V (85－132V) AC 220V (180－260V) 60Hz (50－65Hz)					
Power Consumption		Approx. 70W			Approx. 80W		
Environment	Operating Temperature	0°C－+50°C					
	Storage Temperature	－10°C－+65°C					
	Storage Humidity	20%－80% (Non-Dew)					
Weight		Approx. 10kg net					
External Size		324 (W) ×295 (H) ×387 (D)mm					
Size of Packing	Single carton box	387 (W) ×455 (H) ×397 (D)mm : 12kg					
	Double carton box	459 (W) ×527 (H) ×476 (D)mm : 14kg					
FCC		Class B approved			Class A approved (Class B complied)		



TOEI Toei Electronics Co., Ltd.

特約經銷商

萬達電腦公司

深水埗黃金商場地庫21號

電話：3-610095

* IBM PC、PC/XT is registered trademark of International Business Machine Corp.

** Apple II is a registered trademark of Apple Computer Inc.

學校電腦輔助教育

77 ■ 排序原理透視

BACK UP 資料館

59 ■ LOAD RUNNER MAKER
72 將CP II 4、4C拆成 FILE
70 ■ CRAZY PROTECTION
84 最快的破解 MINDSHADOW

最新遊戲軟件評介

84 FLOPPY
84 RAINY DAY FUN
84 BUCKAROD BANZAR
84 SWASH BUCKLER II

精品店

53 在兼融機上 BOOT PROBYTER
60 80字行兩欄目錄檔案顯示
74 HELLO BREAKER 修改

讀者園地

3 訂閱表[illegible]
2 郵購[illegible]
[illegible] 業餘程式編[illegible]員好消息
51 電腦時代精裝合訂本
48 各期程式[illegible]碟[illegible]目錄
88 AFC通訊
89 一行 兩行程式[illegible]入選作品
87 更正
90 分類廣告

第 22 期電腦時代程式磁碟
可[illegible]列地點購得：
(1) [illegible]／九廣鐵路香港電[illegible]服務站；
(2) 百老滙攝影器材有限公司港九各店；
(3) 各區指定經銷處；
(4) 電腦時代讀者服務部
每張港幣29.90元

■ PROGRAM SAVED ON DISK

特稿

APPLE 開機後的秘密 13
APPLE BOOT ROM剖析 14
■ PRINTSHOP 助手——印製私人名片及信紙 18
■ 活用印字機 22
■ 把機械語化成 BASIC 87

中文咭應用專欄

■ 從程式中學習編寫中文咭程式技巧——砌唐詩遊戲 26

應用軟件庫

■ 唱片柜管理系統 28

中小型工商業應用

■ 電子收銀/計數機 34

小型製作專欄

■ 16K WILDCARD——非常輕易的改良 40

軟件縱橫談

創作冒險遊戲大躍進 42
有放影機功能的 FRAME UP 44

齊齊樂開心坊

■ 麥法基遊戲創作—— 立體死亡大賽車 61
■ 棋類遊戲百種之 6 智力棋的考驗——MASTER MIND 63
■ 德國橋牌 66

專欄

■ PRODOS縱橫 九個工具程式套件 PROBYTER 52
卡通人物專門店 加菲和你說聲HI! 54
麥法基專欄 圖形繪畫法研究(二) 57
機械語言應用 眞正隨機數的產生 60

蘋果工具箱

■ 機械碼輸入錯誤核對工具 74
■ 磁碟壽命性能測試器 75
趨向完美的 DOS——一個獨特的指令 BCOPY 85

電腦時代

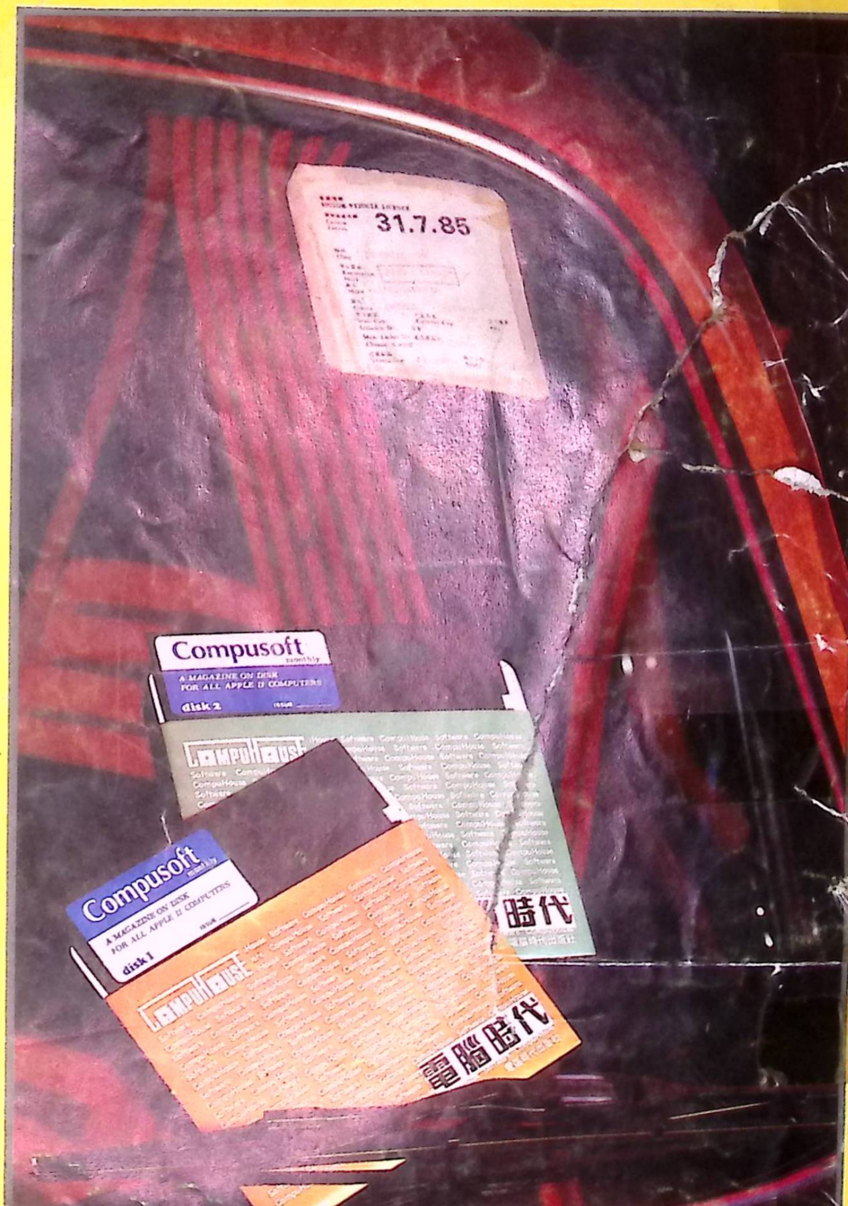
NO. 22

本刊所登載之各個程式全部適用於 APPLE 系列或兼容機種電腦之上

本刊每月廿五號出版。一年十二期。全年訂閱費 120 港元，雜誌連程式磁碟爲 475 港元。海外訂戶請參看訂閱收費表。
■出版：電腦時代出版社 ■督印人：伍小華 ■編輯：電腦時代編輯部 ■軟件支援：COMPUHOUSE ■通訊地址：九龍中央郵箱71193號 ■通訊電話：3-312007 ■ 讀者服務部地址：旺角上海街438號同珍商業中心11字樓1104室（登打士街交界） ■承印者：永恒印刷公司 ■發行人：同德書報社

EVERY MONTH IS DIFFERENT!

COMPUSOFT 磁碟月刊



COMPUSOFT磁碟月刊，每期均出版兩張5 ¼"雙面灌錄的磁碟(合共四面)，以DOS 3.3 FORMAT，適用於任何蘋果電腦或相容的機種。

COMPUSOFT磁碟月刊每期均包羅有極富娛樂、刺激兼並的電腦遊戲。教育軟件、適合任何行業的實用工具程式、電腦自學文章，最新軟件測試報告……等精彩內容，是一本適合電腦發燒友、未來電腦從業員、學校教職員，學生，家長及有志投身電腦軟件工業人士的必備電腦雜誌。

磁碟月刊每期所包含內容的技巧，更是透過一羣專業程序設計師精心泡製的成果，它有着極大的研究價值，是用家們值得珍藏之電腦雜誌。

購備COMPUSOFT磁碟月刊，只花小小代價，獲多多的得益。

每月廿日出版

地鐵／九廣鐵路香港電視服務站特約發售